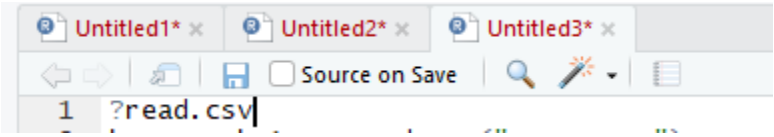
HomeWork 1

Harshith Bondada
M10724854

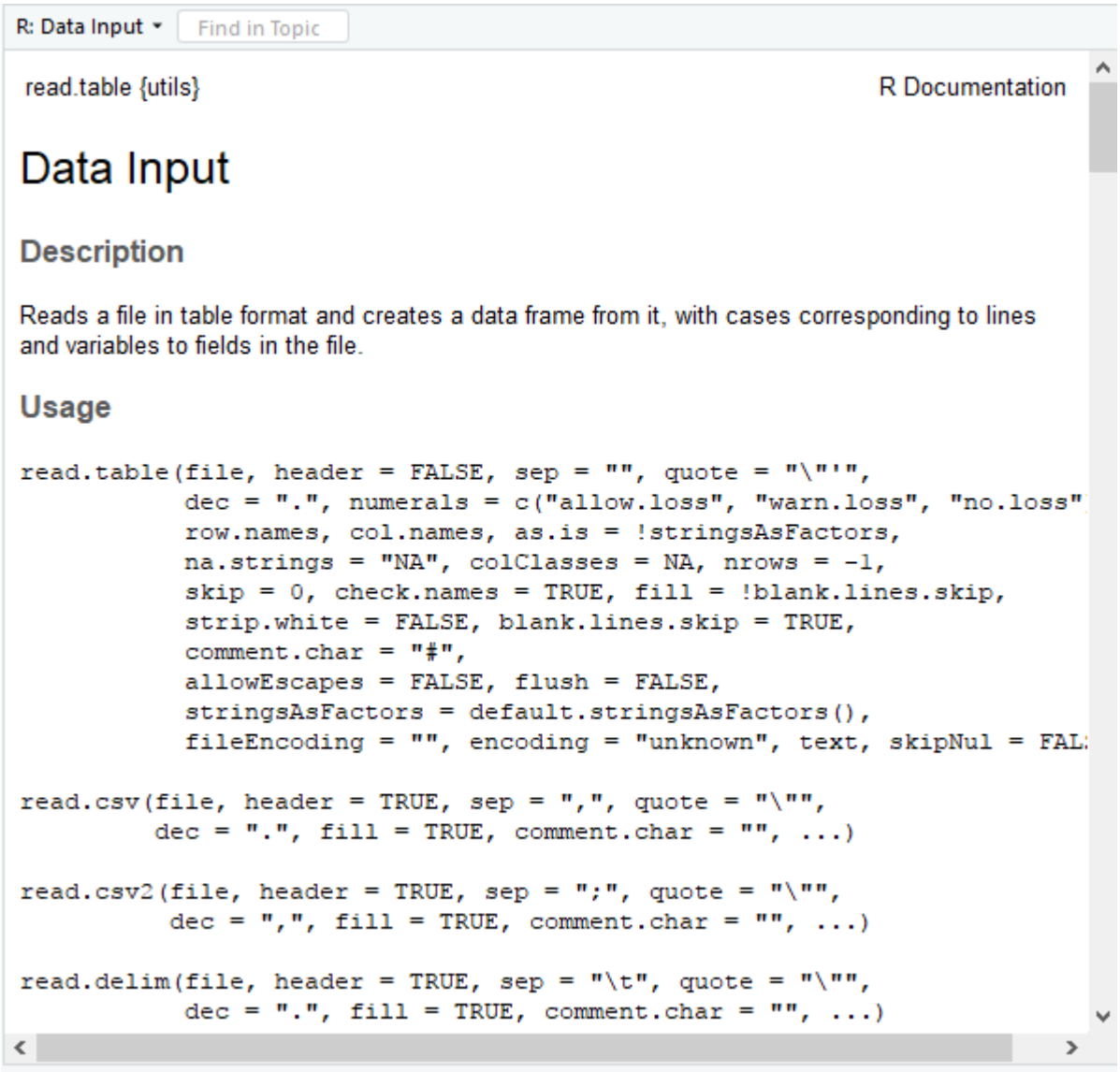
1. Import the CSV file ‘car_r.csv’ using the function “read.table()” or “read.csv()”. Where to find the instruction on how to use the functions?

We use the following command to find the instruction on how to use the function:

? read.csv



We get the documentation for the "read.csv" function as follows:



Harshith Bondada
M10724854

We now use the command to read the data:

home_work_1 <- read.csv("car_r.csv")

```
?read.csv
home_work_1 <- read.csv("car_r.csv")
```

We get the message as follows in the console:

```
[Workspace loaded from ~/.RData]
```

This implies that the data has been successfully loaded into R.

2. How many variables in the data set? What are their names?

We can type in the following commands to determine the number of the variables in the data set.

length(home_work_1)

```
length(home_work_1)
```

We get the following output in the console:

The number of variables in the data set are 9.

```
> length(home_work_1)
[1] 9
```

We can type in the following commands to determine the names of the variables in the data set.

names(home_work_1)

or

ls(home_work_1)

We give the name of the dataset in the names function ie, home_work_1

```
names(home_work_1)
ls(home_work_1)
```

Harshith Bondada
M10724854

We get the following output in console:

The names of the variable are brand, mileage, num_acccidents, num_passengers, speed_car, speed_air, height, width, ABS

```
> names(home_work_1)
[1] "brand"          "mileage"        "num_accidents"  "num_passengers"
[5] "speed_car"      "speed_air"      "height"         "width"
[9] "ABS"
```

3. How many observations in total? How many observations for Ford?

We can type in the following commands to determine the number of the observations in the data set.

Dim is function with name of the dataset as its argument.

dim(home_work_1)

```
  8  dim(home_work_1)
```

We get the following output in the console:

We see that there are 500 observations across 9 variables.

```
> dim(home_work_1)
[1] 500   9
```

We can also use the following command

str(home_work_1)

This returns the structure of the object.

```
 10  str(home_work_1)
```

Harshith Bondada
M10724854

We find the following message in the output console:

```
> str(home_work_1)
'data.frame':   500 obs. of  9 variables:
 $ brand         : Factor w/ 3 levels "Ford","GM","Toyota": 1 3 1 2 1 1 3 1 2 3 ...
 $ mileage       : num  36697 20619 21003 NA 27489 ...
 $ num_accidents : int  2 2 2 0 3 0 2 4 1 3 ...
 $ num_passengers: int  6 14 8 5 6 10 5 5 14 6 ...
 $ speed_car     : num  59.3 38.2 39.2 36.4 49.2 ...
 $ speed_air     : num  -3.2855 -2.7902 -0.2656 -1.3503 -0.0674 ...
 $ height        : num  7.94 6 7.86 5.69 6.05 ...
 $ width         : num  6.07 6.17 6.78 5.85 6.39 ...
 $ ABS           : logi  TRUE TRUE FALSE TRUE FALSE FALSE ...
```

There we can see in the first line that there were 500 observations across 9 variables

For finding the number of observations for Ford, we use the following command:

This command subsets the ford observations into Ford_data.

Here, we are subsetting the ford data by using the **which** function and returning the variables of the brand type ford.

Ford_data=home_work_1[which(home_work_1$brand=='Ford'),]

```
11  Ford_data=home_work_1[ which(home_work_1$brand=='Ford'), ]
```

We get the following data as the output in the console:

```
> dim(Ford_data)
[1] 275   9
```

We find that there are 275 observations across 9 variables for the Ford_data.

Harshith Bondada
M10724854

4. Calculate the mean for each of the car parameters (measures). Please also report the corresponding standard deviation.

We are calculating the mean of car parameters in the below code.

We are using the mean function and mentioning the variable name as its argument.

```
4  mean(mileage ,na.rm = T)
5  mean(num_accidents)
6  mean(num_passengers)
7  mean(speed_air)
8  mean(speed_car)
9  mean(height)
10 mean(width)
11 sd(mileage, na.rm= T)
12 sd(num_accidents)
13 sd(num_passengers)
14 sd(speed_air)
15 sd(speed_car)
16 sd(height)
17 sd(width)
```

Harshith Bondada
M10724854

We are getting the output for the car parameters in the below pasted output screen.

```
> mean(mileage ,na.rm = T)
[1] 39564.63
> mean(mileage ,na.rm = T)
[1] 39564.63
> mean(num_accidents)
[1] 2.154
> mean(num_passengers)
[1] 6.69
> mean(speed_air)
[1] 0.245482
> mean(speed_car)
[1] 50.05964
> mean(height)
[1] 5.914164
> mean(width)
[1] 6.013667
> sd(mileage, na.rm= T)
[1] 10819.68
> sd(num_accidents)
[1] 1.423495
> sd(num_passengers)
[1] 3.742983
> sd(speed_air)
[1] 3.084353
> sd(speed_car)
[1] 9.77354
> sd(height)
[1] 1.054882
> sd(width)
[1] 0.4714572
>
```

We now calculate the mean of each parameters based on the car brand.

We can type in the following command to find the mean for each of the car parameters:

mean(mileage[brand=="Ford"],na.rm = T)

This command will take the variable mileage for the Ford car and calculates the Mean.

Harshith Bondada
M10724854

In this command we select the the Ford car by mention brand=="Ford", this returns the mean for only Ford car.

Similarly, we mention the brand types for GM and Toyota and return their respective mean.

```
mean(mileage[brand=="Ford"],na.rm = T)
mean(num_accidents[brand=="Ford"])
mean(num_passengers[brand=="Ford"])
mean(speed_car[brand=="Ford"])
mean(speed_air[brand=="Ford"])
mean(height[brand=="Ford"])
mean(width[brand=="Ford"])
mean(ABS[brand=="Ford"])
table(brand)
mean(mileage[brand=="GM"],na.rm = T)
mean(num_accidents[brand=="GM"])
mean(num_passengers[brand=="GM"])
mean(speed_car[brand=="GM"])
mean(speed_air[brand=="GM"])
mean(height[brand=="GM"])
mean(width[brand=="GM"])
mean(ABS[brand=="GM"])
mean(mileage[brand=="Toyota"],na.rm = T)
mean(num_accidents[brand=="Toyota"])
mean(num_passengers[brand=="Toyota"])
mean(speed_car[brand=="Toyota"])
mean(speed_air[brand=="Toyota"])
mean(height[brand=="Toyota"])
mean(width[brand=="Toyota"])
mean(ABS[brand=="Toyota"])
```

The output we get in the console is as follows:

```
> mean(mileage[brand=="Ford"],na.rm = T)
[1] 39876.61
> mean(num_accidents[brand=="Ford"])
[1] 2.269091
> mean(num_passengers[brand=="Ford"])
[1] 6.509091
> mean(speed_car[brand=="Ford"])
[1] 49.96355
> mean(speed_air[brand=="Ford"])
[1] 0.3343119
> mean(height[brand=="Ford"])
[1] 5.941855
> mean(width[brand=="Ford"])
[1] 6.064743
> mean(ABS[brand=="Ford"])
[1] 0.6
```

Harshith Bondada
M10724854

```
> mean(mileage[brand=="GM"],na.rm = T)
[1] 40397.95
> mean(num_accidents[brand=="GM"])
[1] 1.98
> mean(num_passengers[brand=="GM"])
[1] 7.22
> mean(speed_car[brand=="GM"])
[1] 50.20667
> mean(speed_air[brand=="GM"])
[1] 0.2262234
> mean(height[brand=="GM"])
[1] 5.9406
> mean(width[brand=="GM"])
[1] 5.906073
> mean(ABS[brand=="GM"])
[1] 0.56
> mean(mileage[brand=="Toyota"],na.rm = T)
[1] 38229.74
> mean(num_accidents[brand=="Toyota"])
[1] 2.04
> mean(num_passengers[brand=="Toyota"])
[1] 6.664
> mean(speed_car[brand=="Toyota"])
[1] 50.15341
> mean(speed_air[brand=="Toyota"])
[1] 0.06546285
> mean(height[brand=="Toyota"])
[1] 5.832094
> mean(width[brand=="Toyota"])
[1] 5.987375
> mean(ABS[brand=="Toyota"])
[1] 0.632
>
```

Harshith Bondada
M10724854

Calculating the SD:

This is same as we did for the mean function. Here we use the sd function and return the sd values of the variables based on the brand of the car.

```
57  sd(mileage[brand=="Ford"],na.rm = T)
58  sd(num_accidents[brand=="Ford"])
59  sd(num_passengers[brand=="Ford"])
60  sd(speed_car[brand=="Ford"])
61  sd(speed_air[brand=="Ford"])
62  sd(height[brand=="Ford"])
63  sd(width[brand=="Ford"])
64  sd(ABS[brand=="Ford"])
65  sd(mileage[brand=="GM"],na.rm = T)
66  sd(num_accidents[brand=="GM"])
67  sd(num_passengers[brand=="GM"])
68  sd(speed_car[brand=="GM"])
69  sd(speed_air[brand=="GM"])
70  sd(height[brand=="GM"])
71  sd(width[brand=="GM"])
72  sd(ABS[brand=="GM"])
73  sd(mileage[brand=="Toyota"],na.rm = T)
74  sd(num_accidents[brand=="Toyota"])
75  sd(num_passengers[brand=="Toyota"])
76  sd(speed_car[brand=="Toyota"])
77  sd(speed_air[brand=="Toyota"])
78  sd(height[brand=="Toyota"])
79  sd(width[brand=="Toyota"])
80  sd(ABS[brand=="Toyota"])
```

Harshith Bondada
M10724854

The output in the console is as follows:

We can see the sd's of each car parameter's based on their brand type.

```
> sd(mileage[brand=="Ford"],na.rm = T)
[1] 11006.35
> sd(num_accidents[brand=="Ford"])
[1] 1.429725
> sd(num_passengers[brand=="Ford"])
[1] 3.543642
> sd(speed_car[brand=="Ford"])
[1] 9.583966
> sd(speed_air[brand=="Ford"])
[1] 2.962368
> sd(height[brand=="Ford"])
[1] 1.025994
> sd(width[brand=="Ford"])
[1] 0.4712387
> sd(ABS[brand=="Ford"])
[1] 0.4907911
> sd(mileage[brand=="GM"],na.rm = T)
[1] 10542.86
> sd(num_accidents[brand=="GM"])
[1] 1.385203
> sd(num_passengers[brand=="GM"])
[1] 4.405185
> sd(speed_car[brand=="GM"])
[1] 9.938125
> sd(speed_air[brand=="GM"])
[1] 3.344762
> sd(height[brand=="GM"])
[1] 1.094198
> sd(width[brand=="GM"])
[1] 0.4765816
> sd(ABS[brand=="GM"])
[1] 0.4988877
> sd(mileage[brand=="Toyota"],na.rm = T)
[1] 10644.94
> sd(num_accidents[brand=="Toyota"])
[1] 1.427834
> sd(num_passengers[brand=="Toyota"])
```

Harshith Bondada
M10724854

```
> sd(num_passengers[brand=="Toyota"])
[1] 3.582988
> sd(speed_car[brand=="Toyota"])
[1] 10.12594
> sd(speed_air[brand=="Toyota"])
[1] 3.147925
> sd(height[brand=="Toyota"])
[1] 1.089516
> sd(width[brand=="Toyota"])
[1] 0.4545493
> sd(ABS[brand=="Toyota"])
[1] 0.484202
>
```

Harshith Bondada
M10724854

5. Obtain the histogram for each of the car parameters.

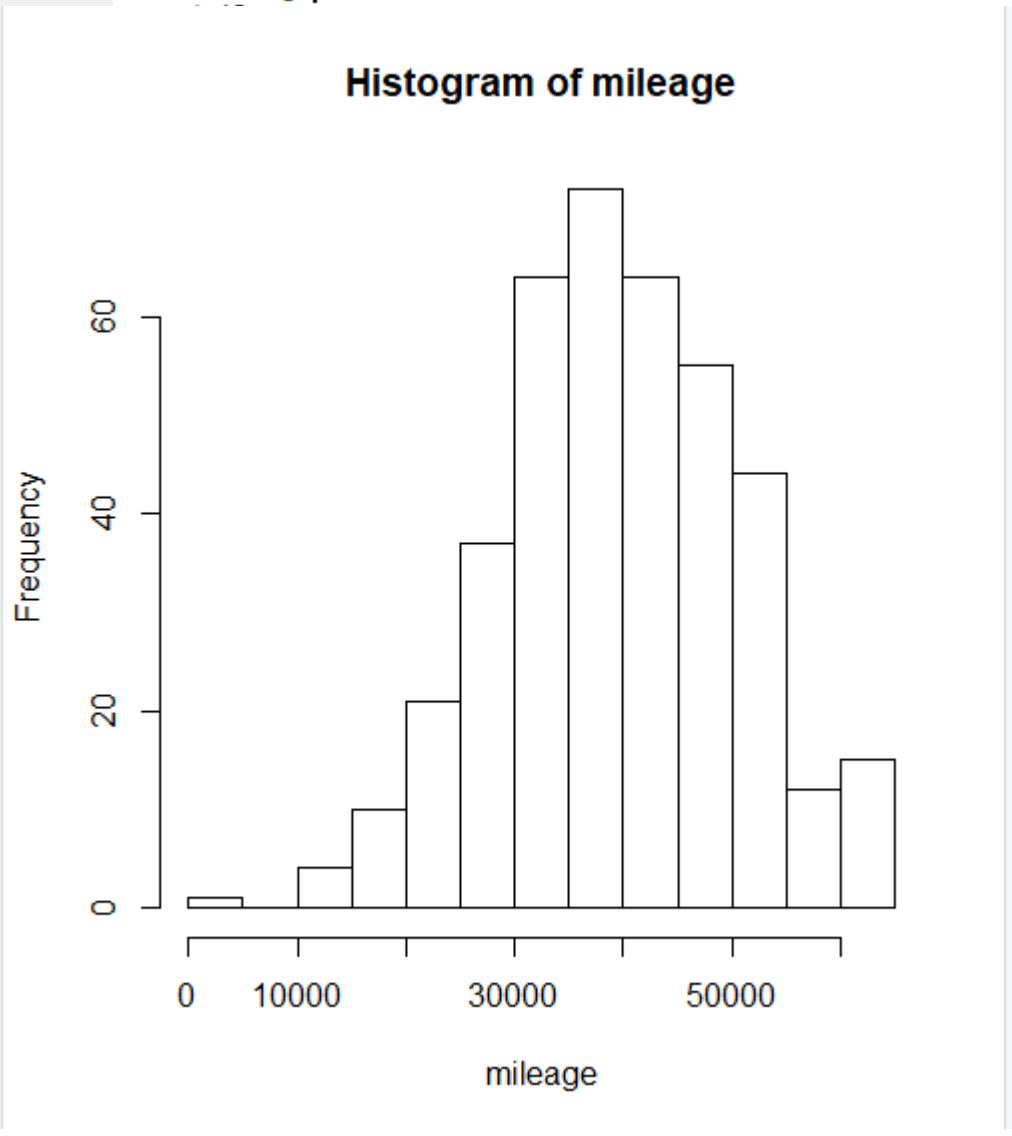
I have calculated the histogram for each car parameters as well as parameters based on the car brand.

First we now go with car parameters:

This screenshot represents the histogram for mileage.

We use the hist function.

```
4 hist(mileage)
```

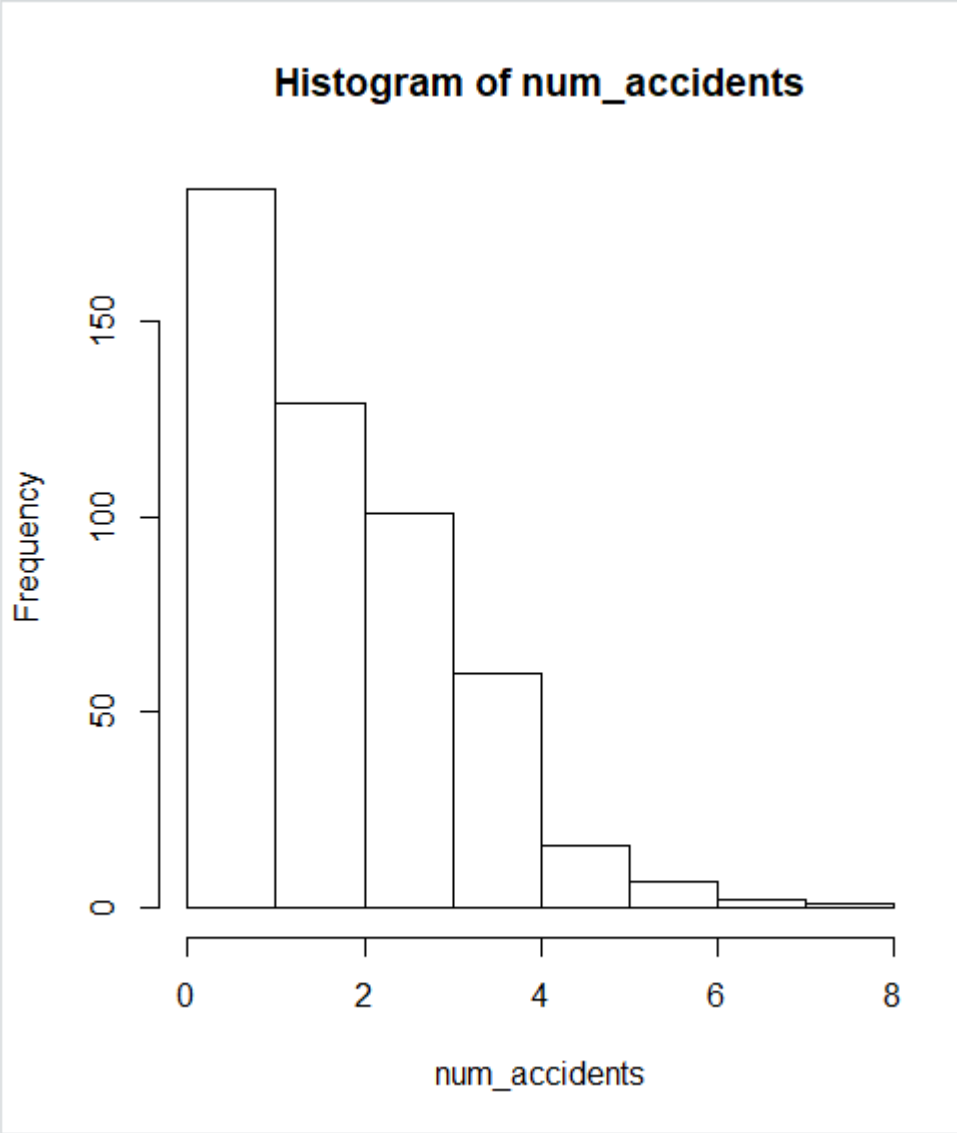


Harshith Bondada
M10724854

This screenshot represents the histogram for num_accidents.

We use the hist function.

```
4  hist(num_accidents)
```

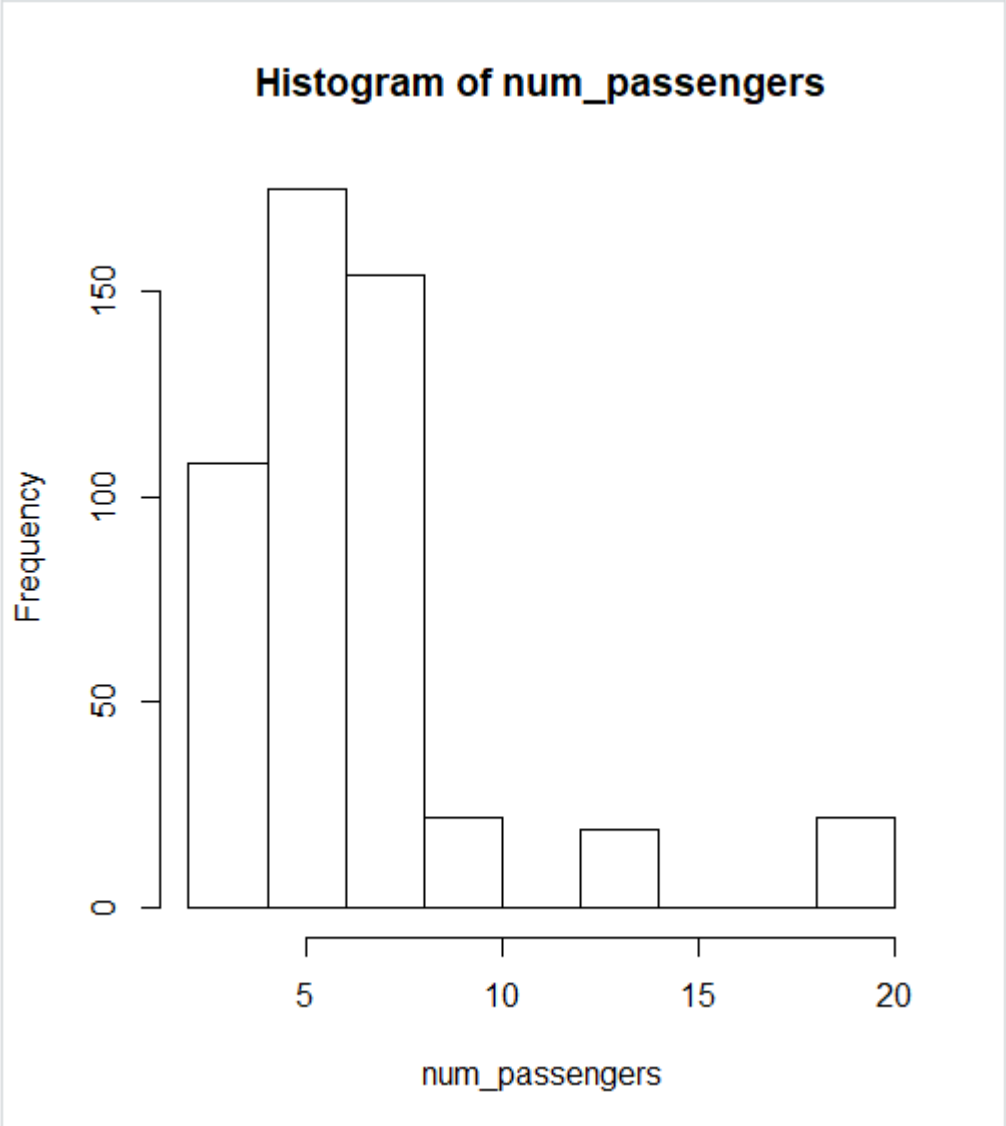


Harshith Bondada
M10724854

This screenshot represents the histogram for num_passengers.

We use the hist function.

```
4 hist(num_passengers)
```

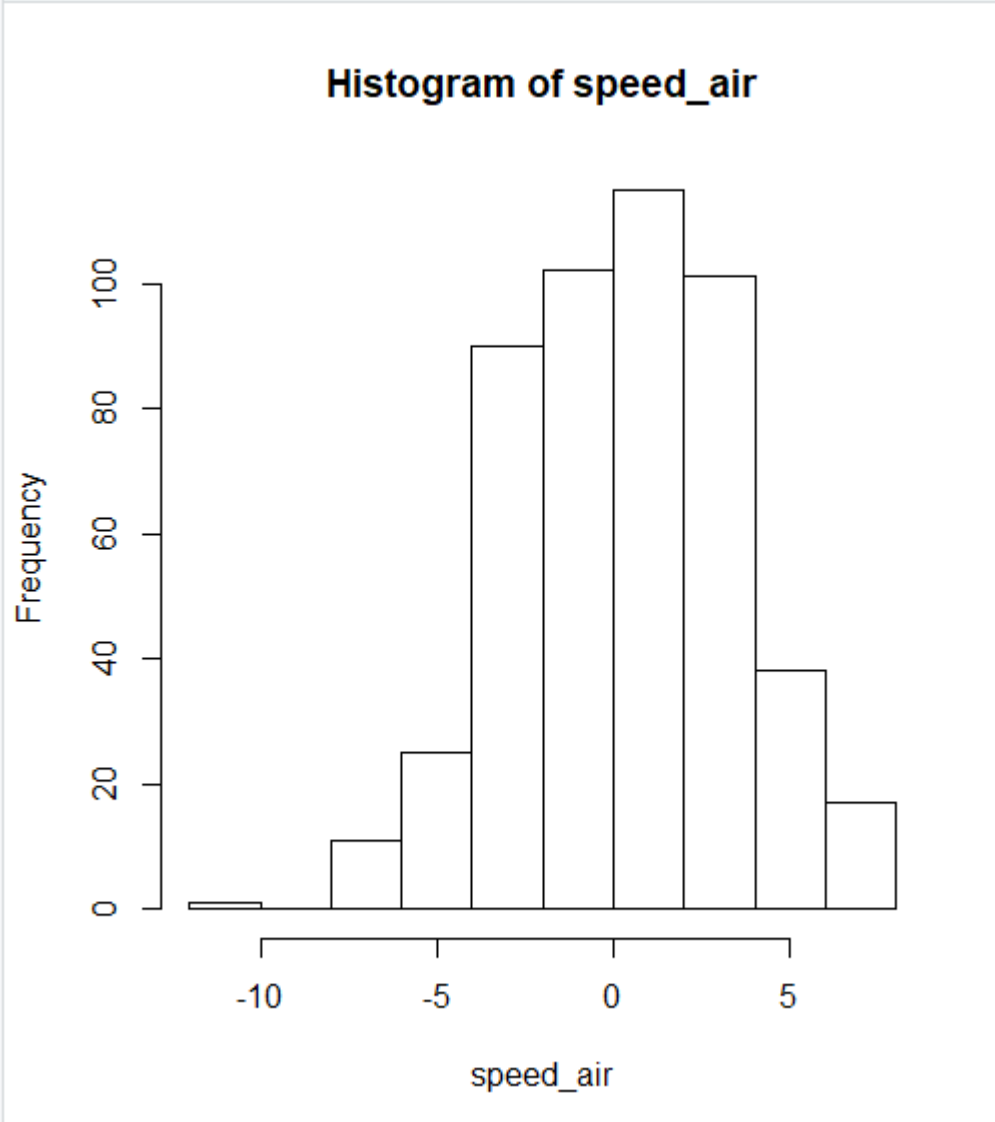


Harshith Bondada
M10724854

This screenshot represents the histogram for speed_air.

We use the hist function.

```
4 hist(speed_air)
```

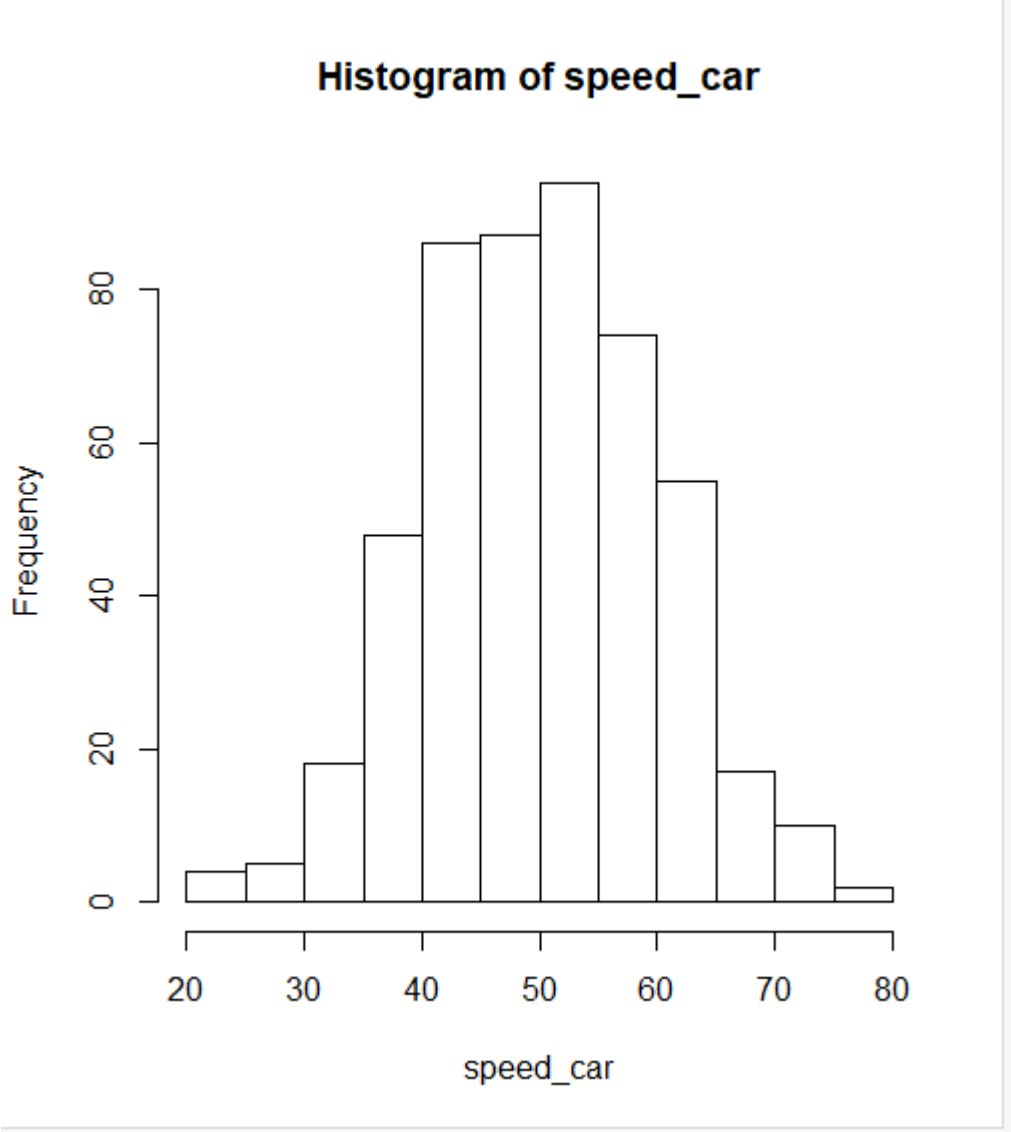


Harshith Bondada
M10724854

This screenshot represents the histogram for speed_car.

We use the hist function.

```
4  hist(speed_car)
```

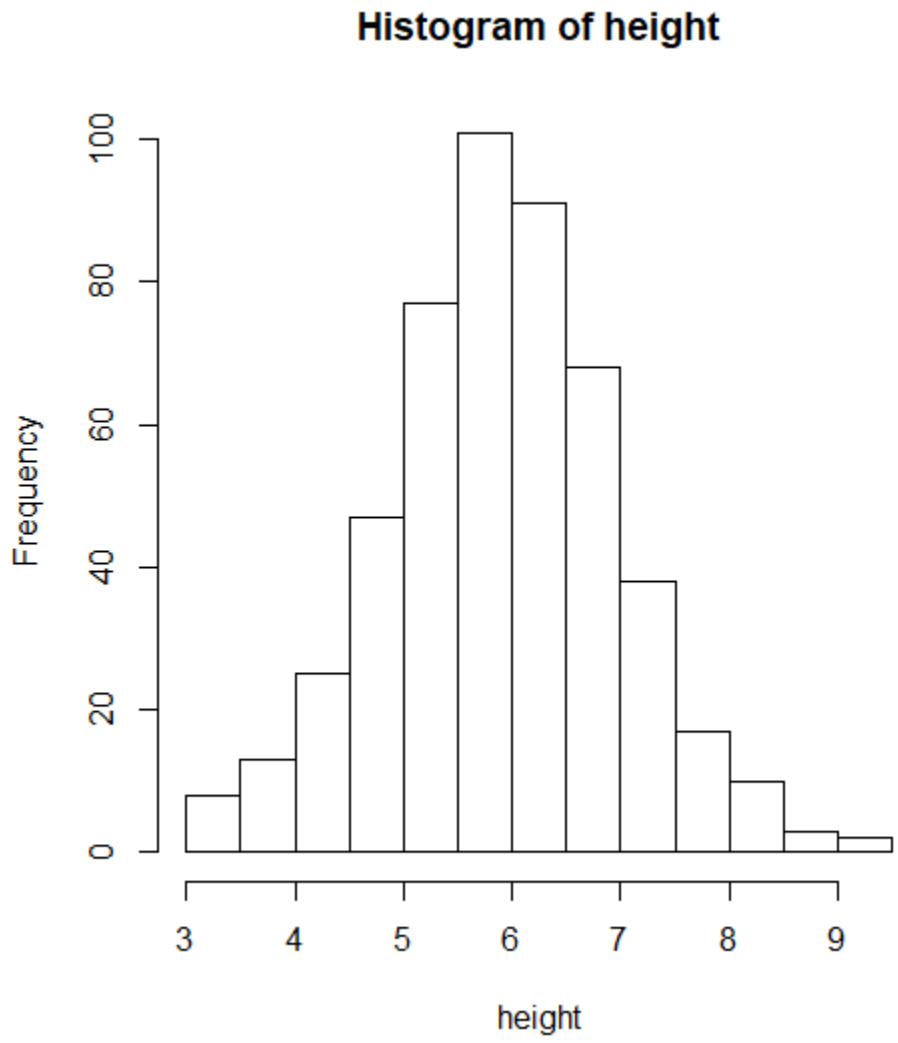


Harshith Bondada
M10724854

This screenshot represents the histogram for height

We use the hist function.

```
4 hist(height)
```

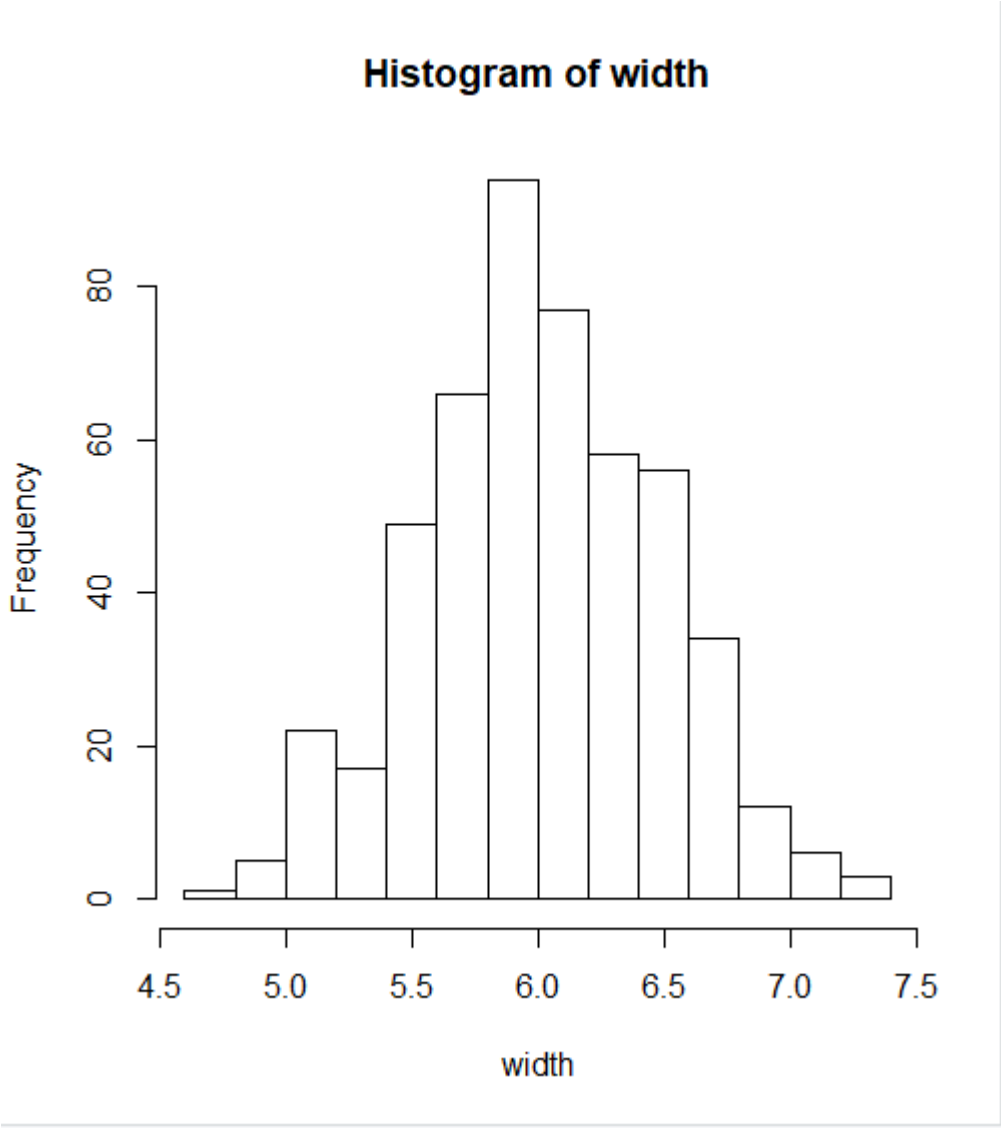


Harshith Bondada
M10724854

This screenshot represents the histogram for width.

We use the hist function.

```
4 hist(width)
```



Harshith Bondada
M10724854

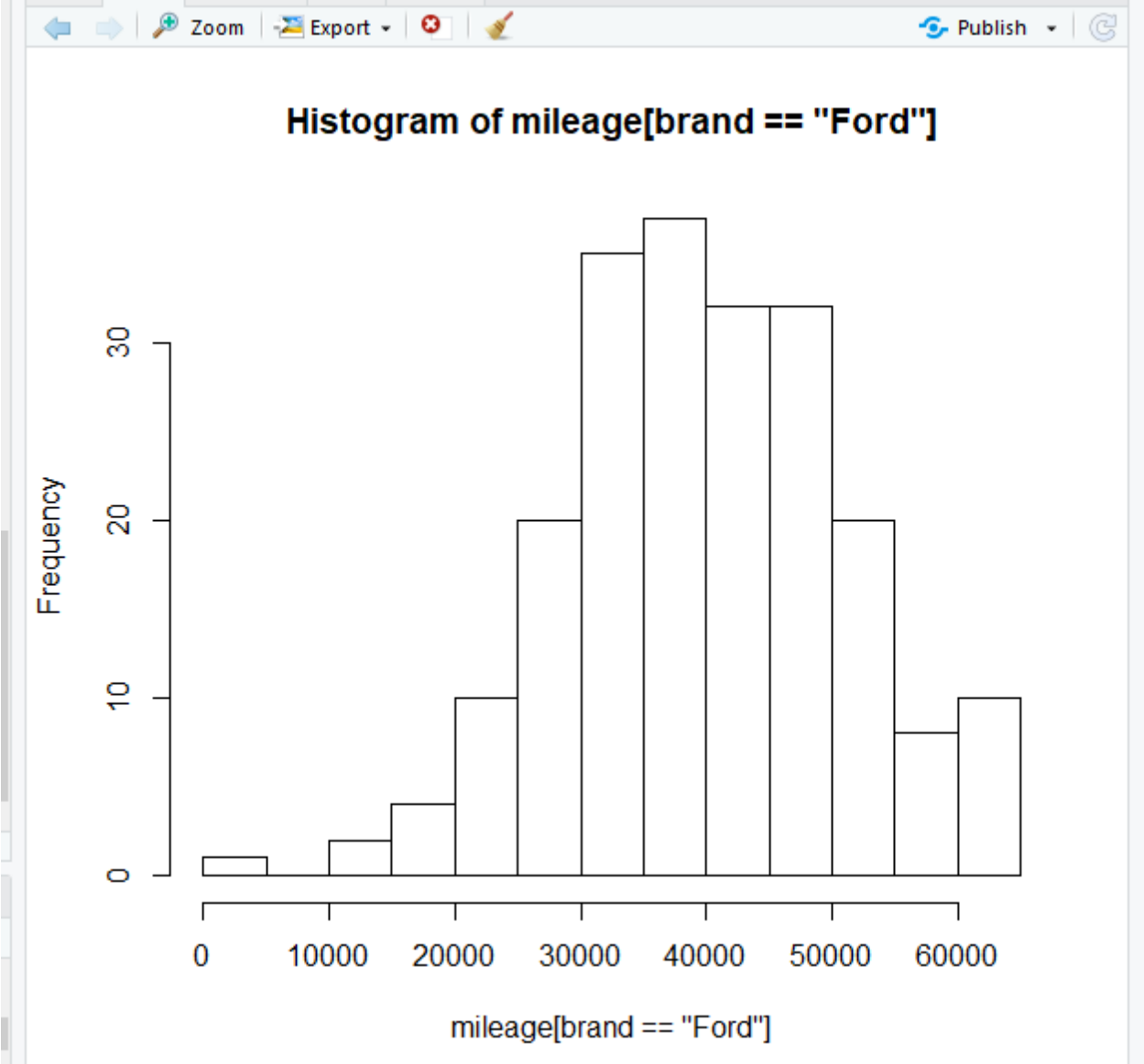
We now calculate the histogram for each car parameter based on its brand

The command we use for obtaining the histogram for each of the car parameters is as follows:

We are obtaining the histogram for each car parameters based on their brand type.

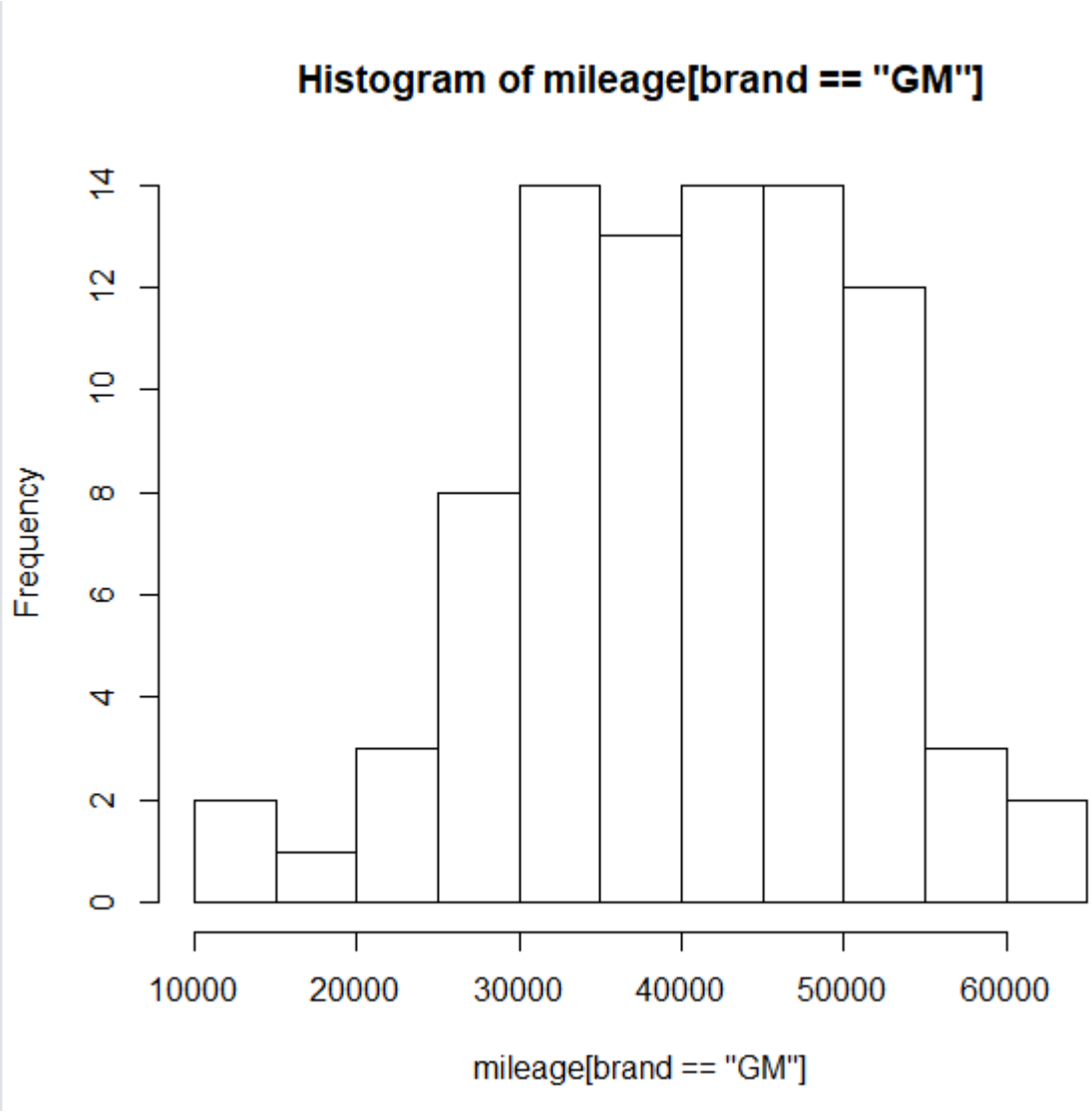
This below fig represents the histogram for mileage for Ford Car.

```
81  hist(mileage[brand=="Ford"])
```



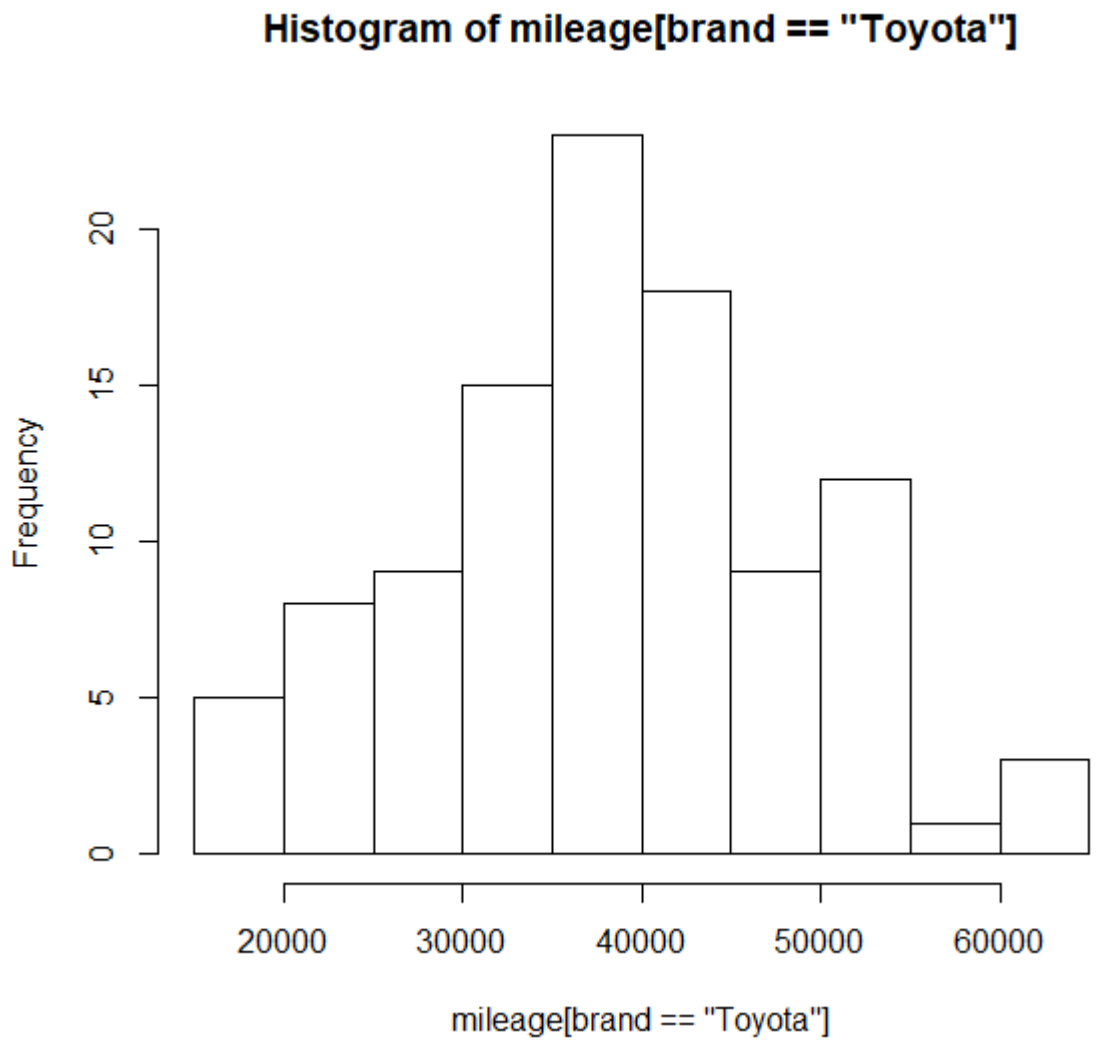
This represents the histogram mileage for GM car.

```
81  hist(mileage[brand=="GM"])
```



This represents the histogram mileage for Toyota car.

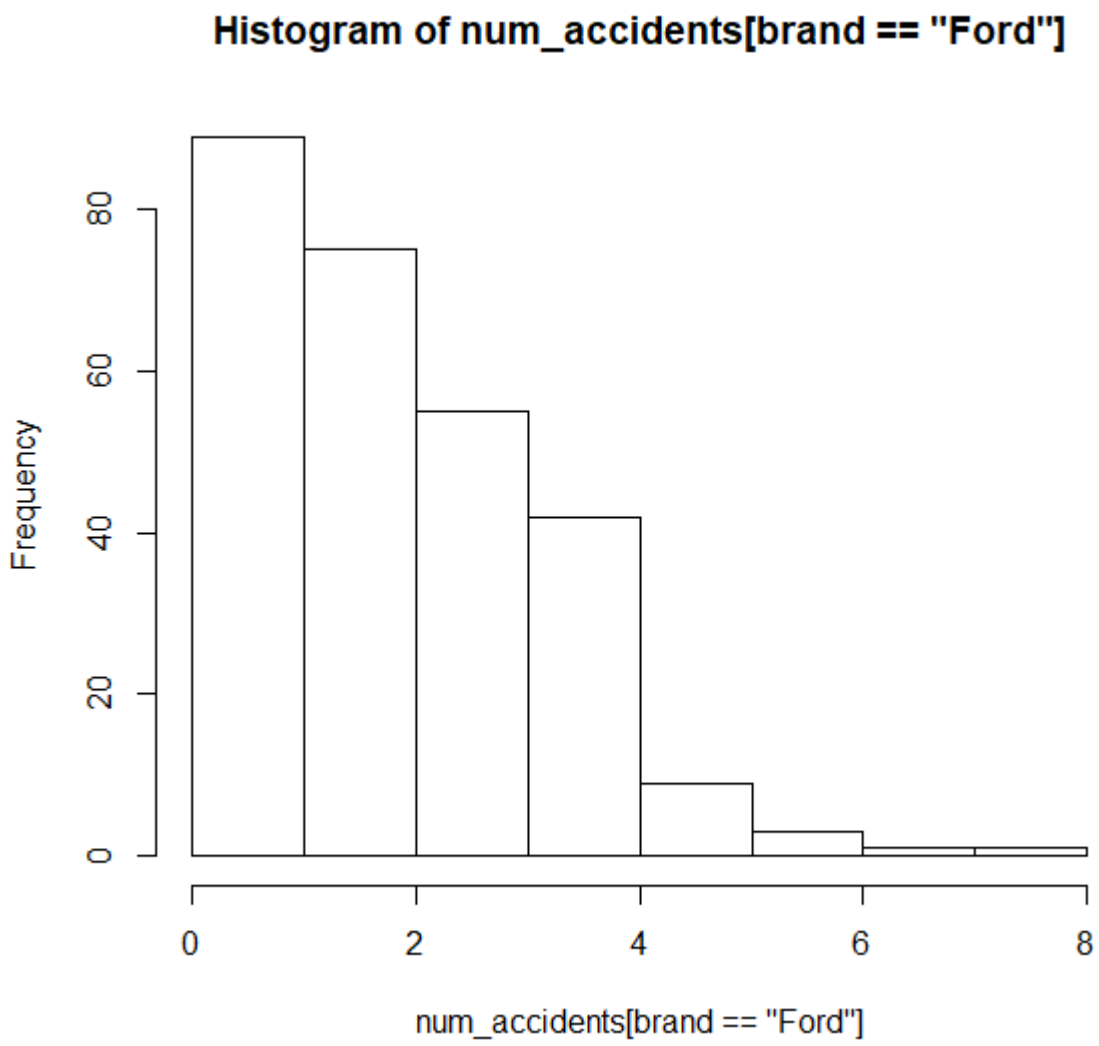
```
81  hist(mileage[brand=="Toyota"])
```



Harshith Bondada
M10724854

This represents the histogram for num_accidents for Ford car.

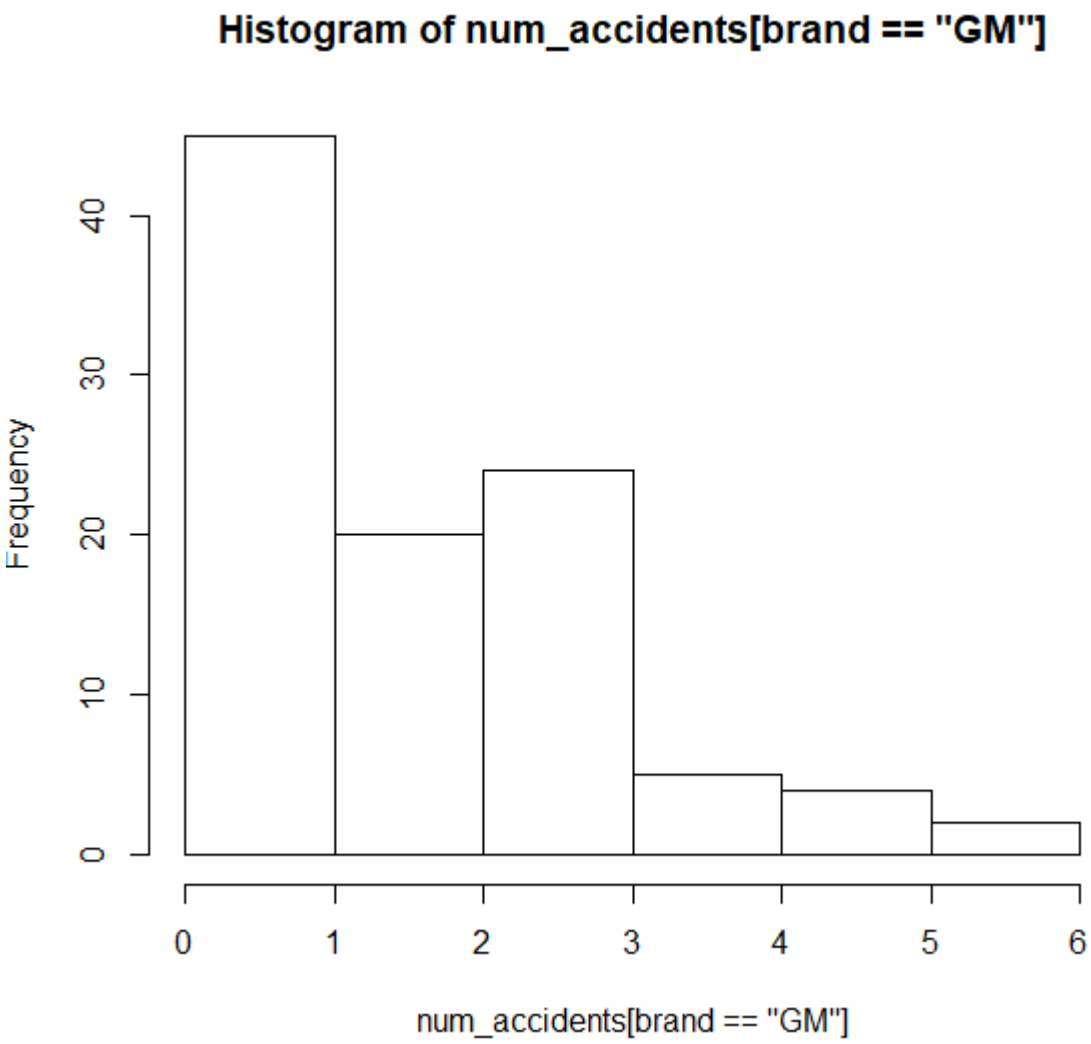
```
81  hist(num_accidents[brand=="Ford"])
```



Harshith Bondada
M10724854

This represents the histogram for num_accidents for GM brand car

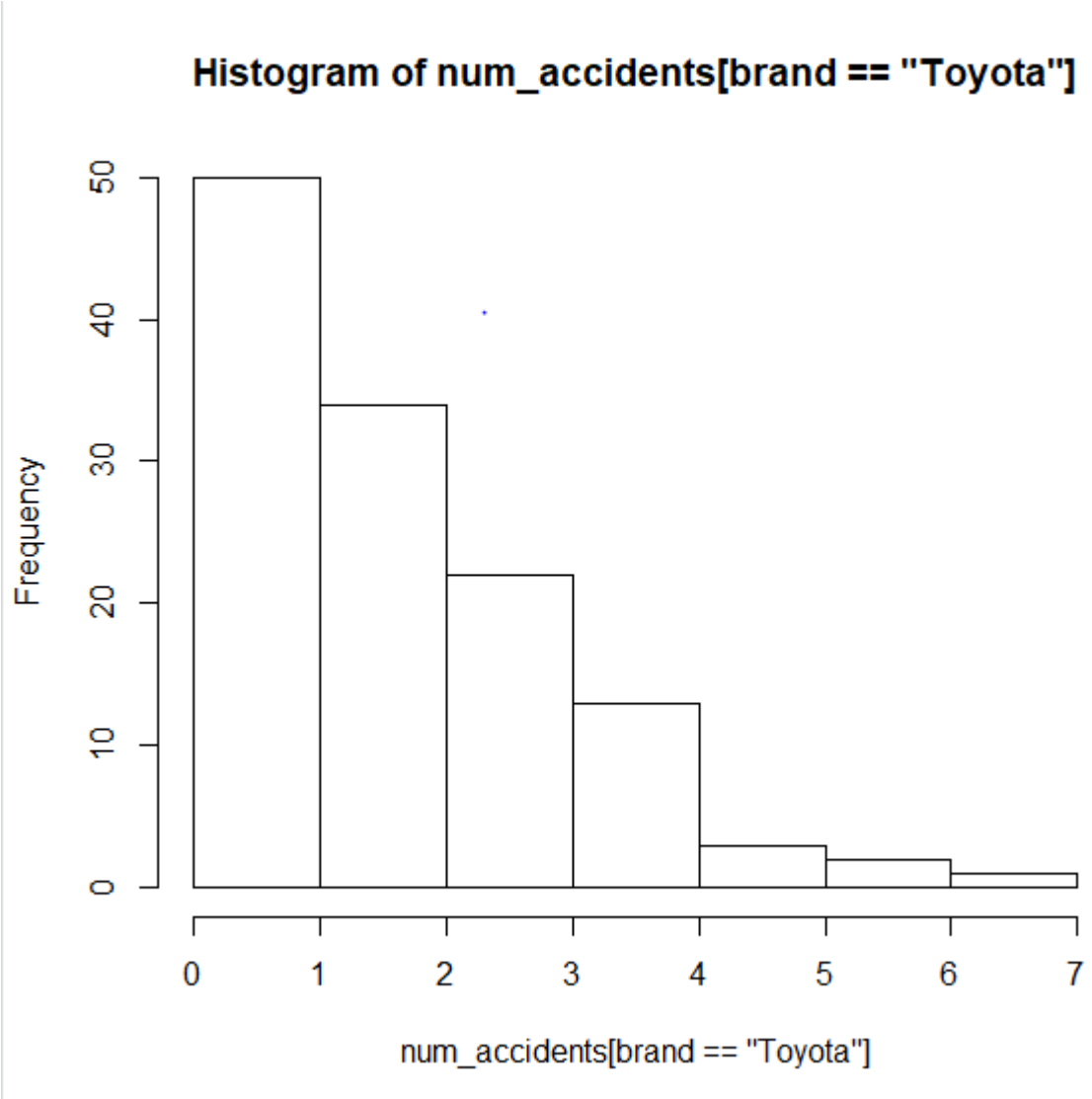
```
81  hist(num_accidents[brand=="GM"])
```



Harshith Bondada
M10724854

This represents the histogram for num_accidents for Toyota brand car

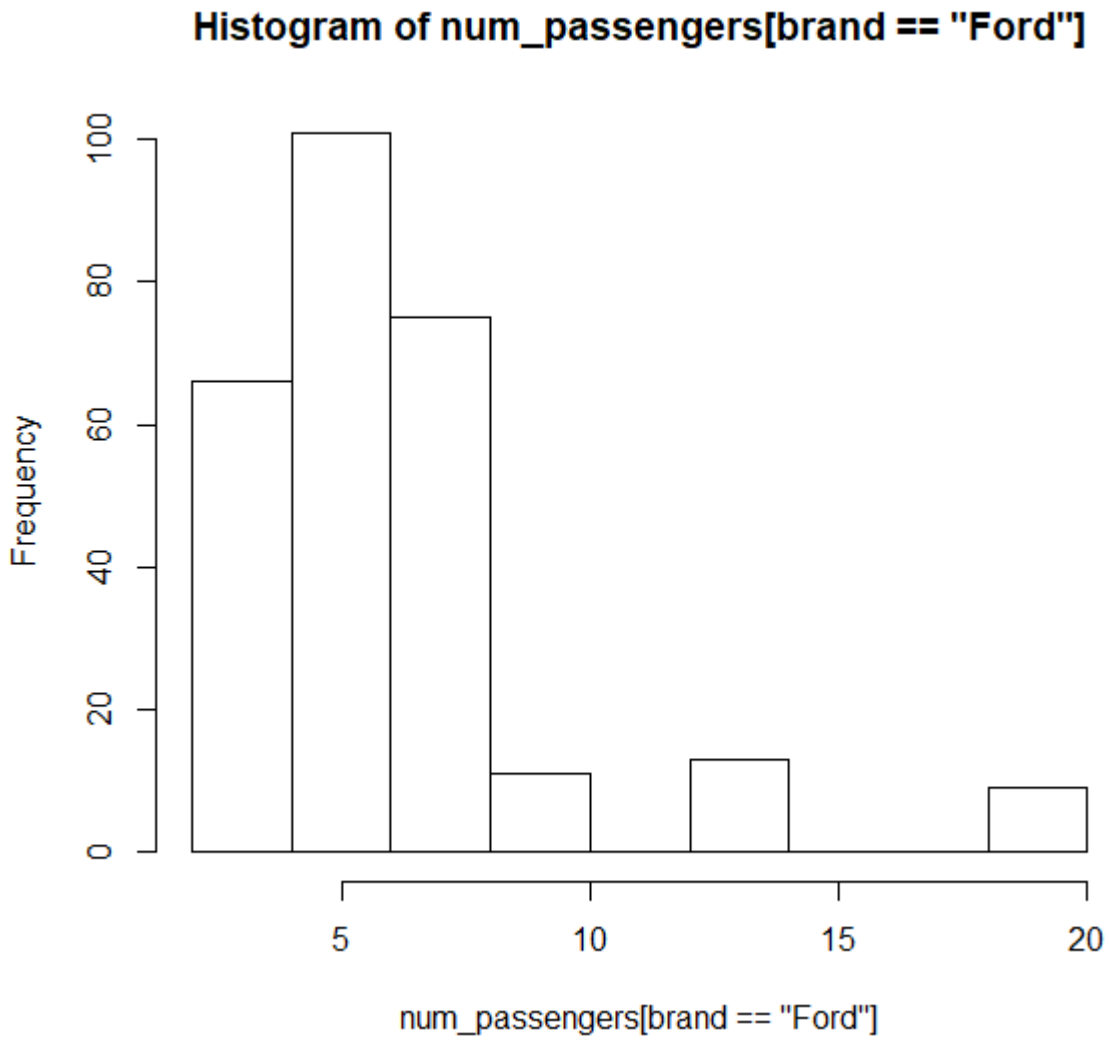
```
81  hist(num_accidents[brand=="Toyota"])
```



Harshith Bondada
M10724854

This represents the histogram for num_passengers for Ford brand car

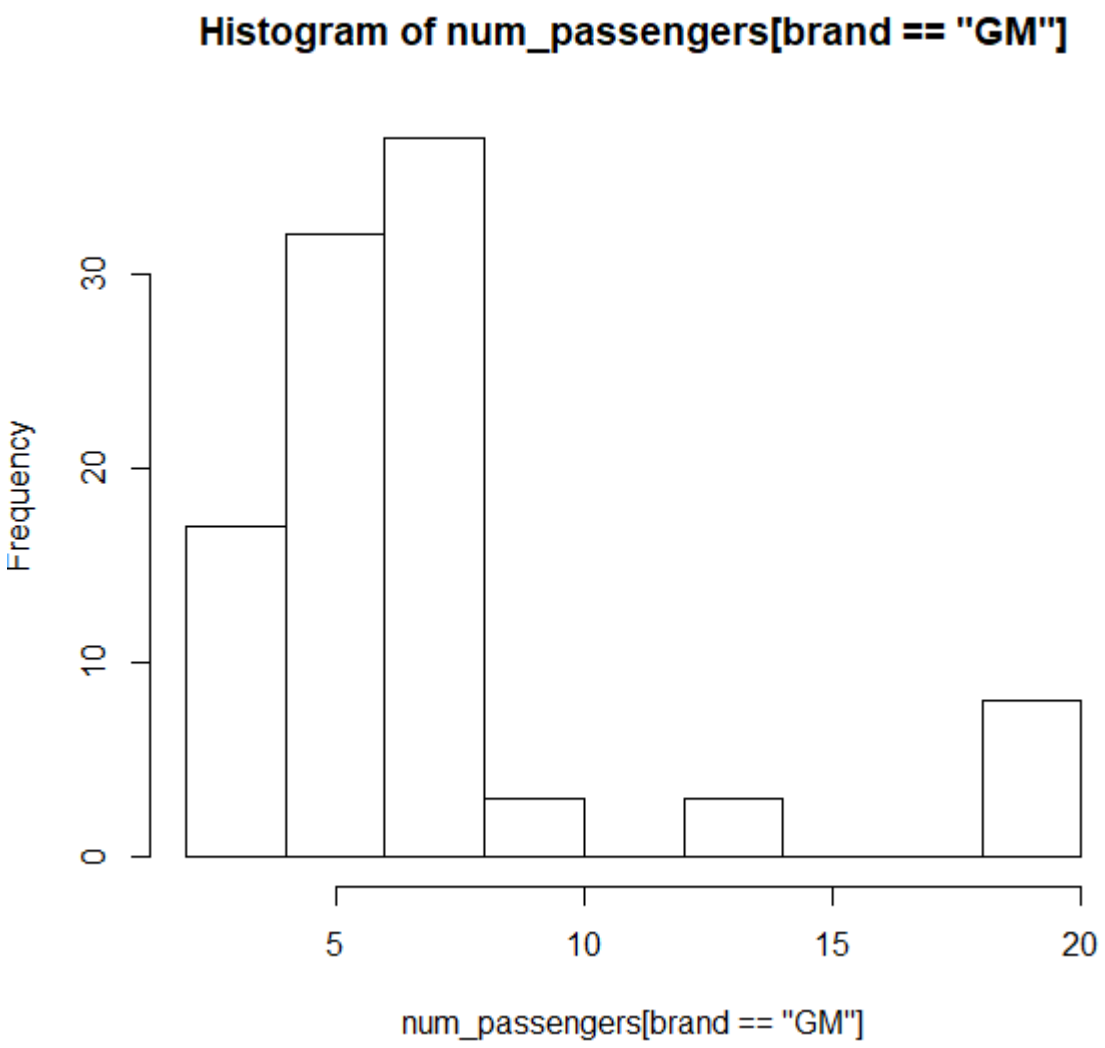
```
81  hist(num_passengers[brand=="Ford"])
```



Harshith Bondada
M10724854

This represents the histogram for num_passengers for GM brand car

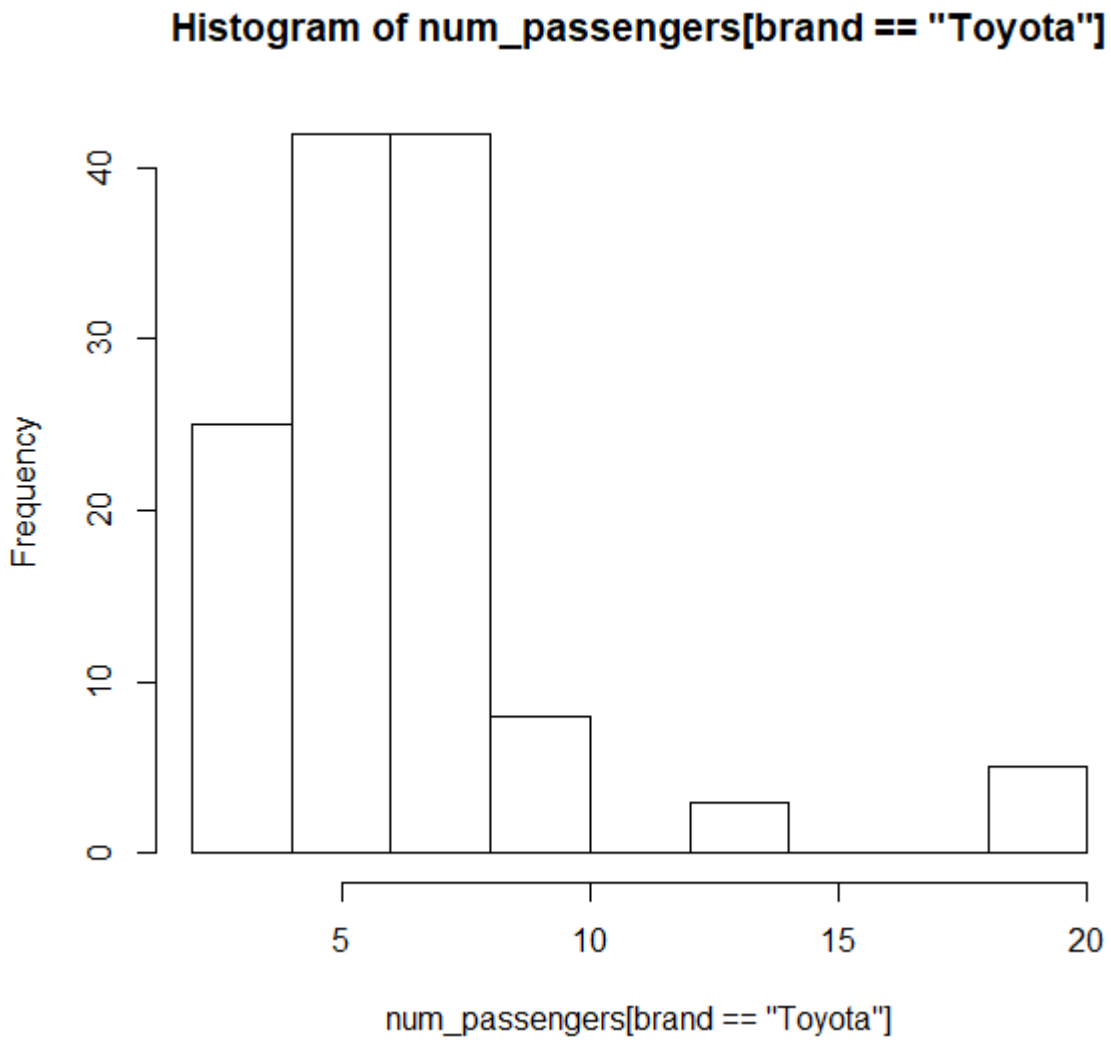
```
81  hist(num_passengers[brand=="GM"])
```



Harshith Bondada
M10724854

This represents the histogram for num_passengers for Toyota brand car

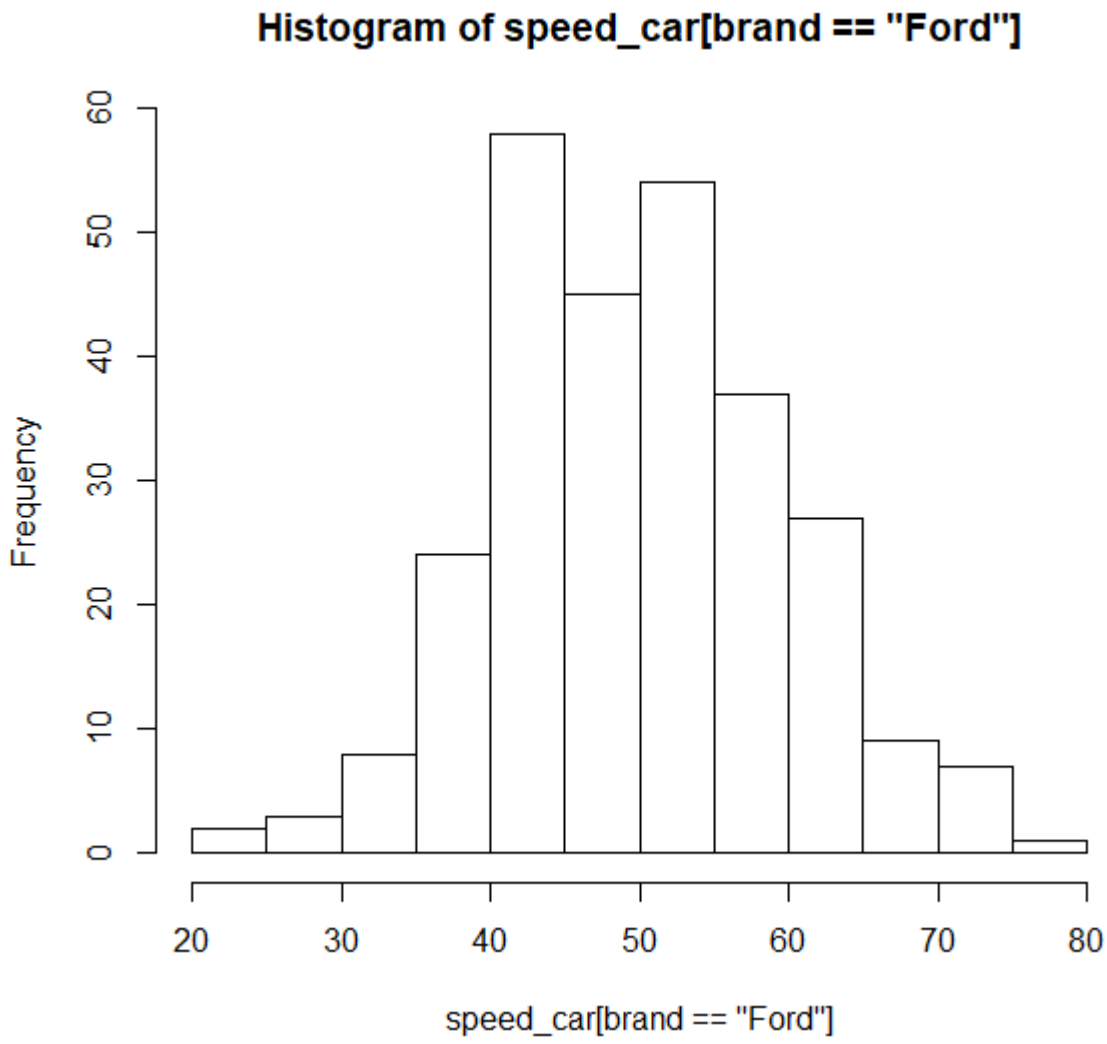
```
81  hist(num_passengers[brand=="Toyota"])
```



Harshith Bondada
M10724854

This represents the histogram for speed_car for Ford brand car

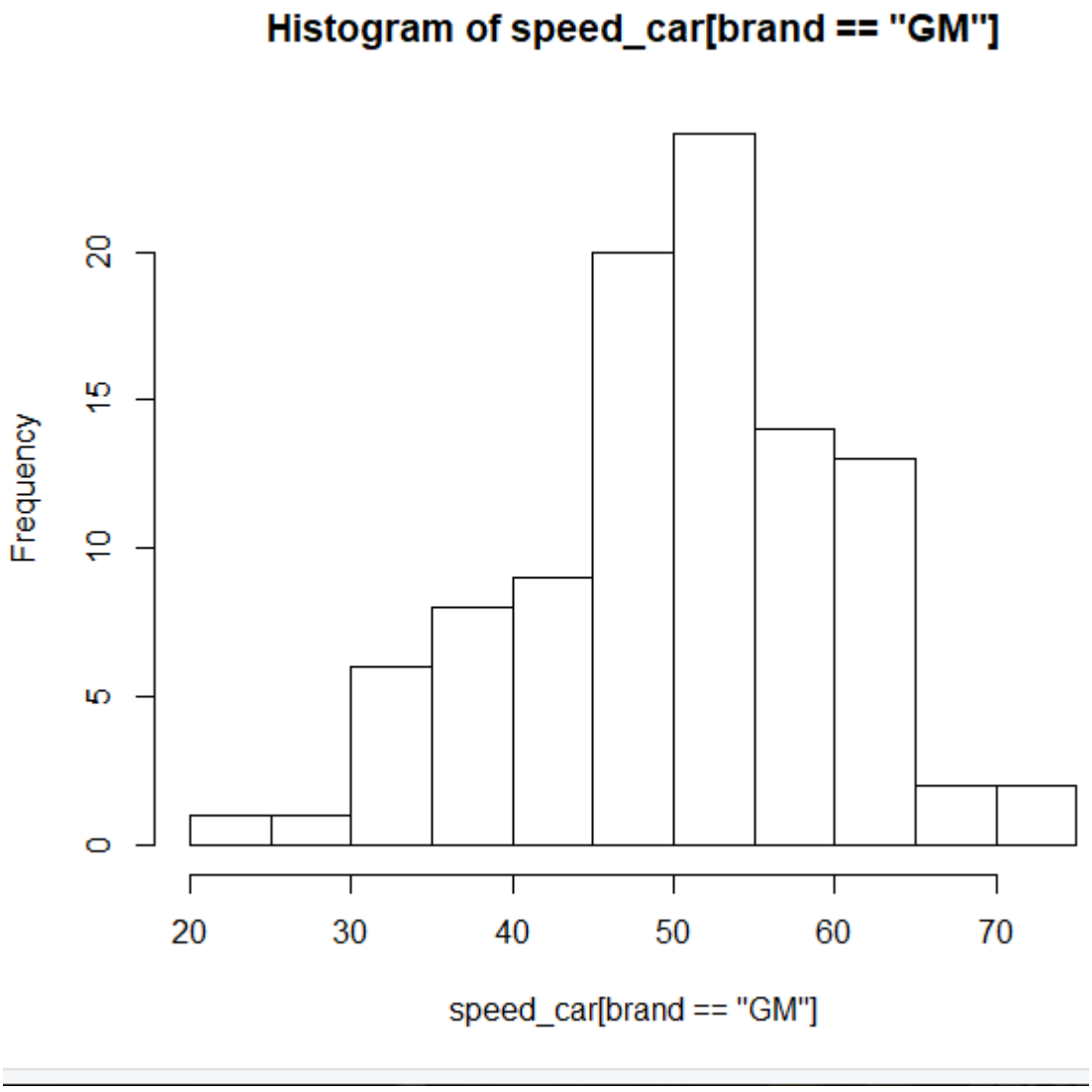
```
81  hist(speed_car[brand=="Ford"])
```



Harshith Bondada
M10724854

This represents the histogram for speed_car for GM brand car

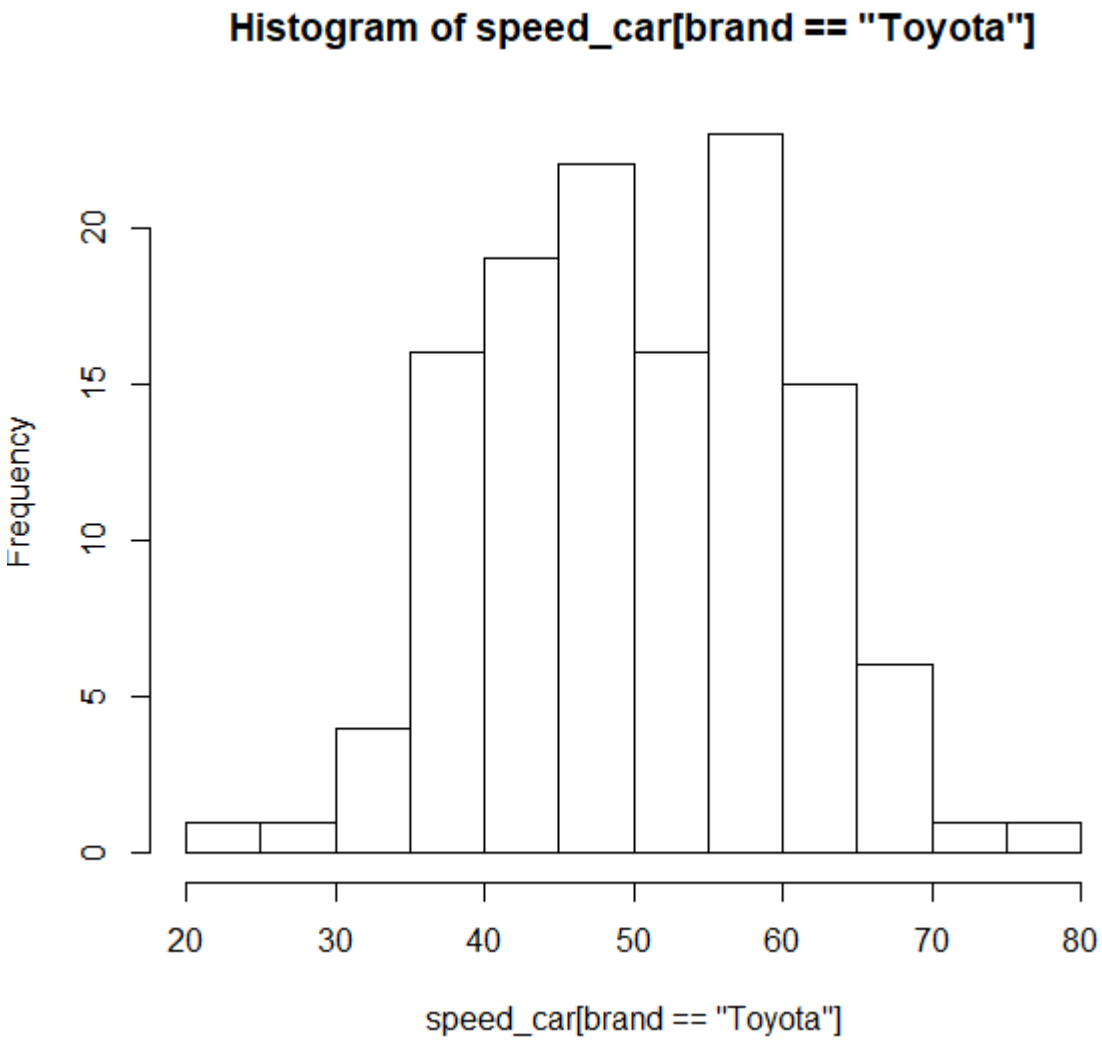
```
81  hist(speed_car[brand=="GM"])
```



Harshith Bondada
M10724854

This represents the histogram for speed_car for Toyota brand car

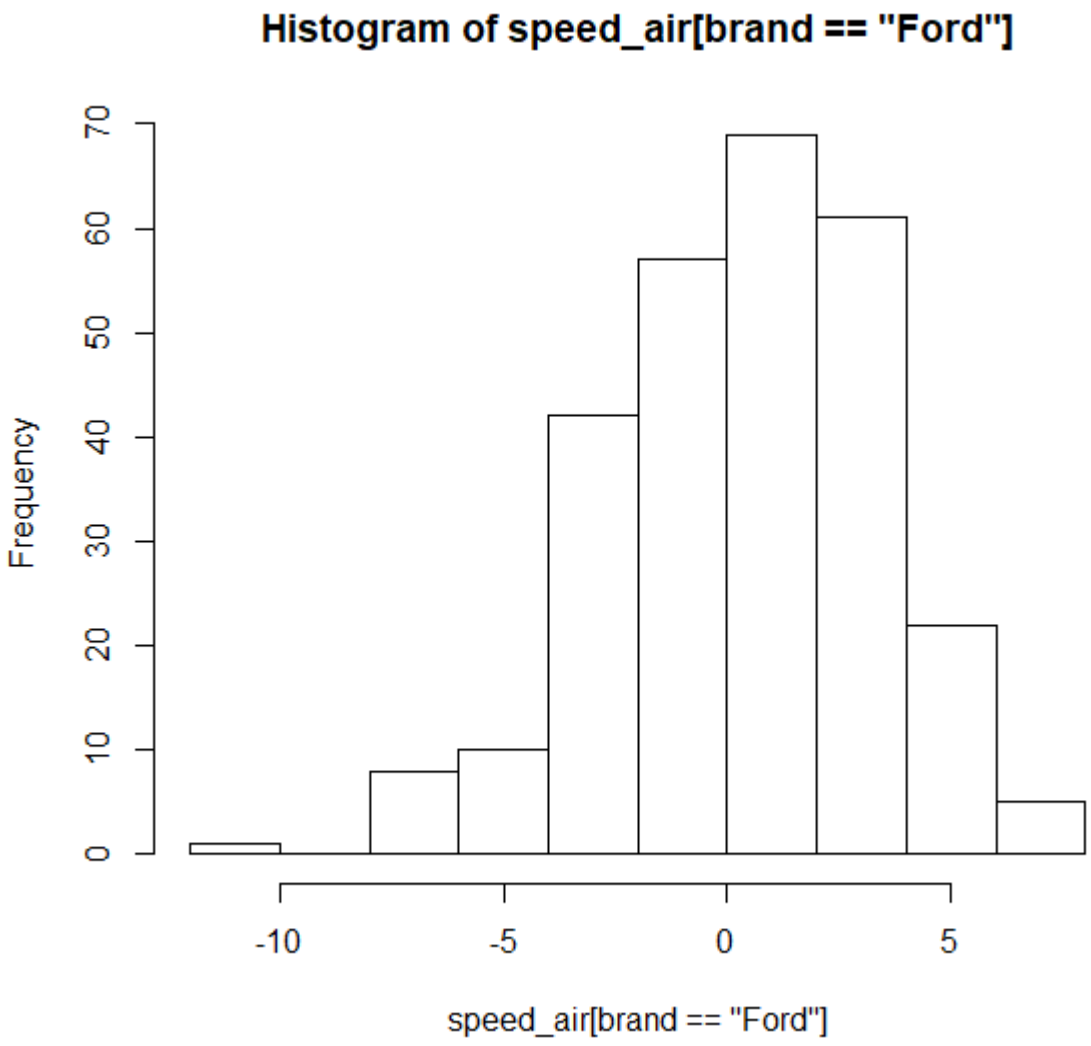
```
81  hist(speed_car[brand=="Toyota"])
```



Harshith Bondada
M10724854

This represents the histogram for speed_air for Ford brand car.

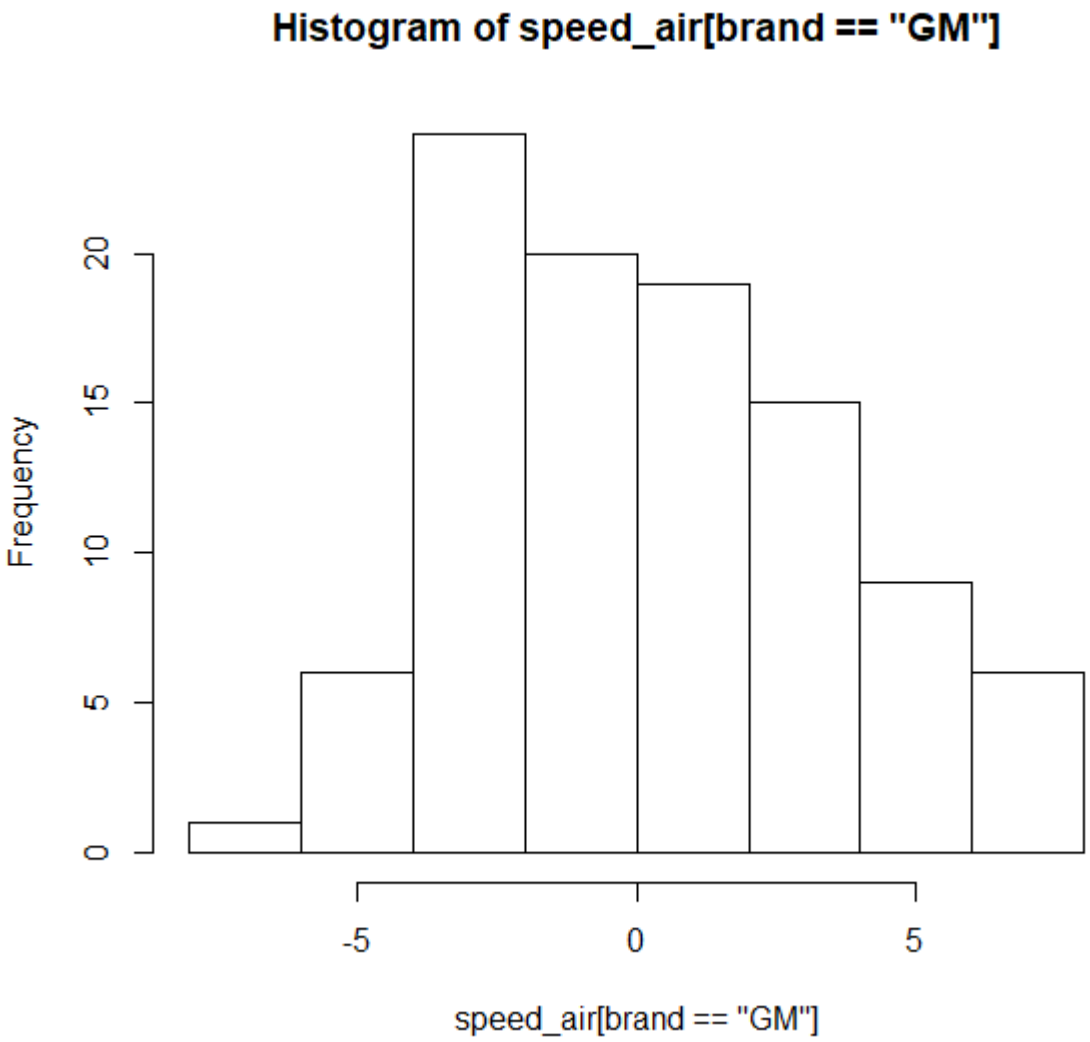
```
81  hist(speed_air[brand=="Ford"])
```



Harshith Bondada
M10724854

This represents the histogram for speed_air for GM brand car.

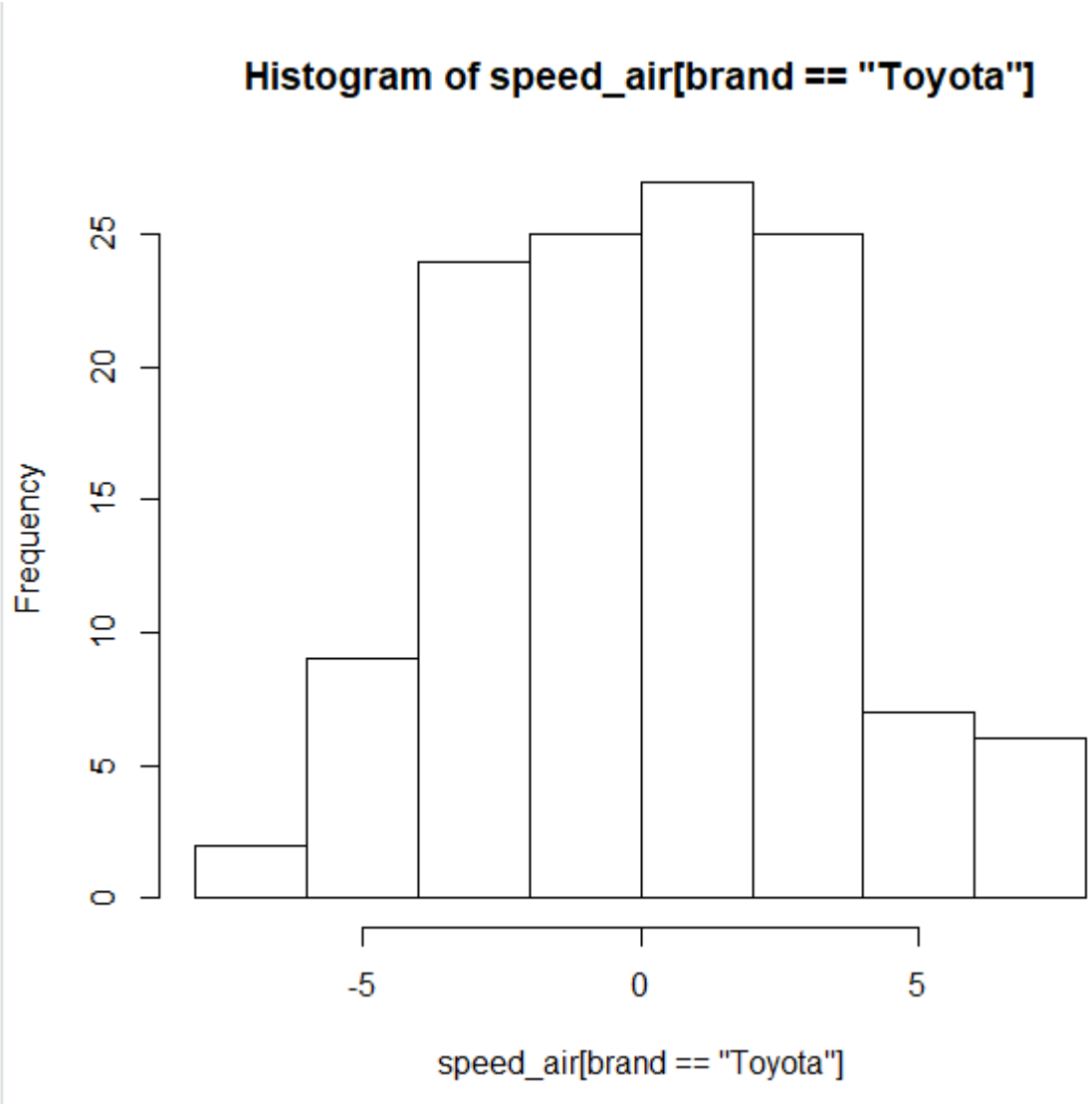
```
81  hist(speed_air[brand=="GM"])
```



Harshith Bondada
M10724854

This represents the histogram for speed_air for Toyota brand car.

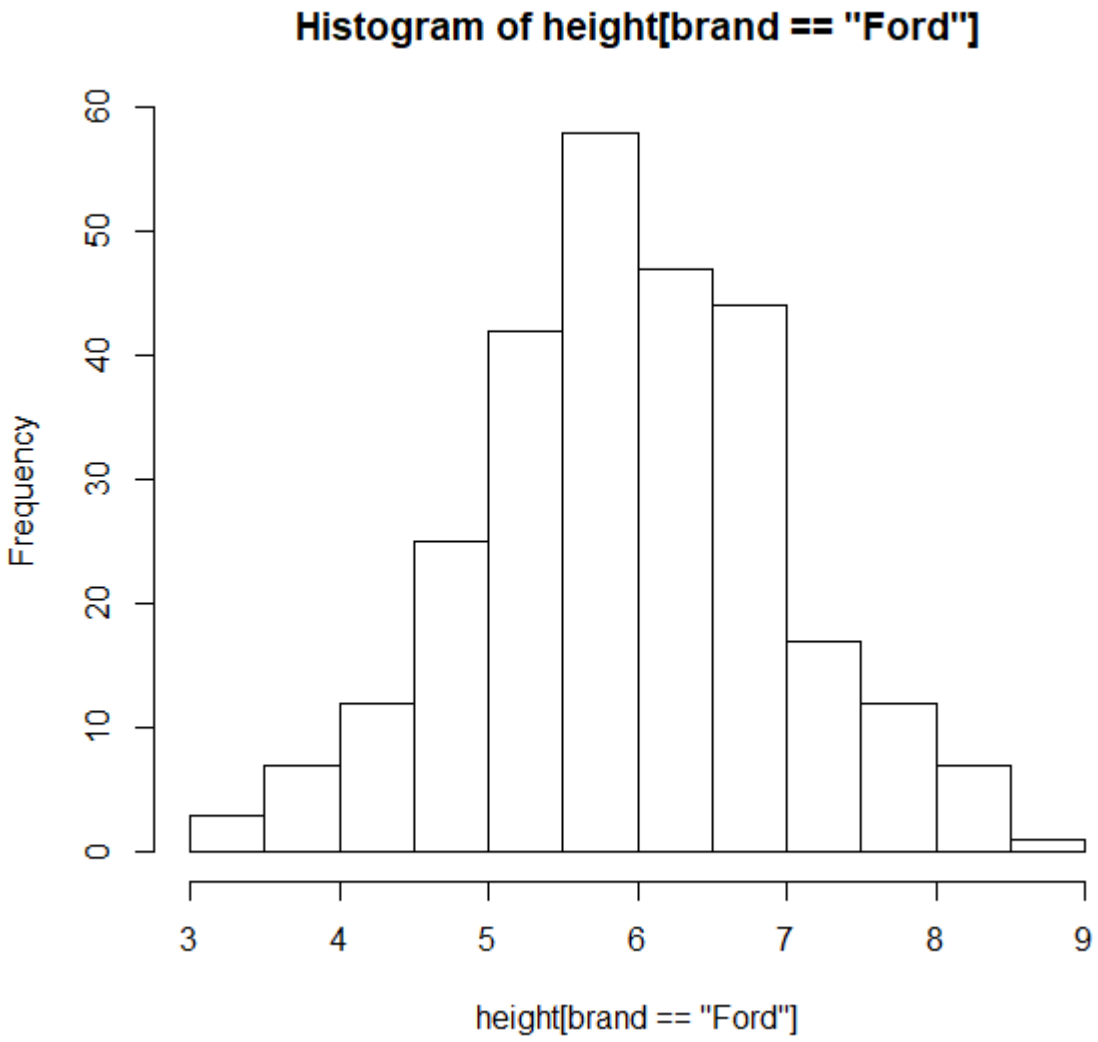
```
81  hist(speed_air[brand=="Toyota"])
```



Harshith Bondada
M10724854

This represents the histogram for height for Ford brand car.

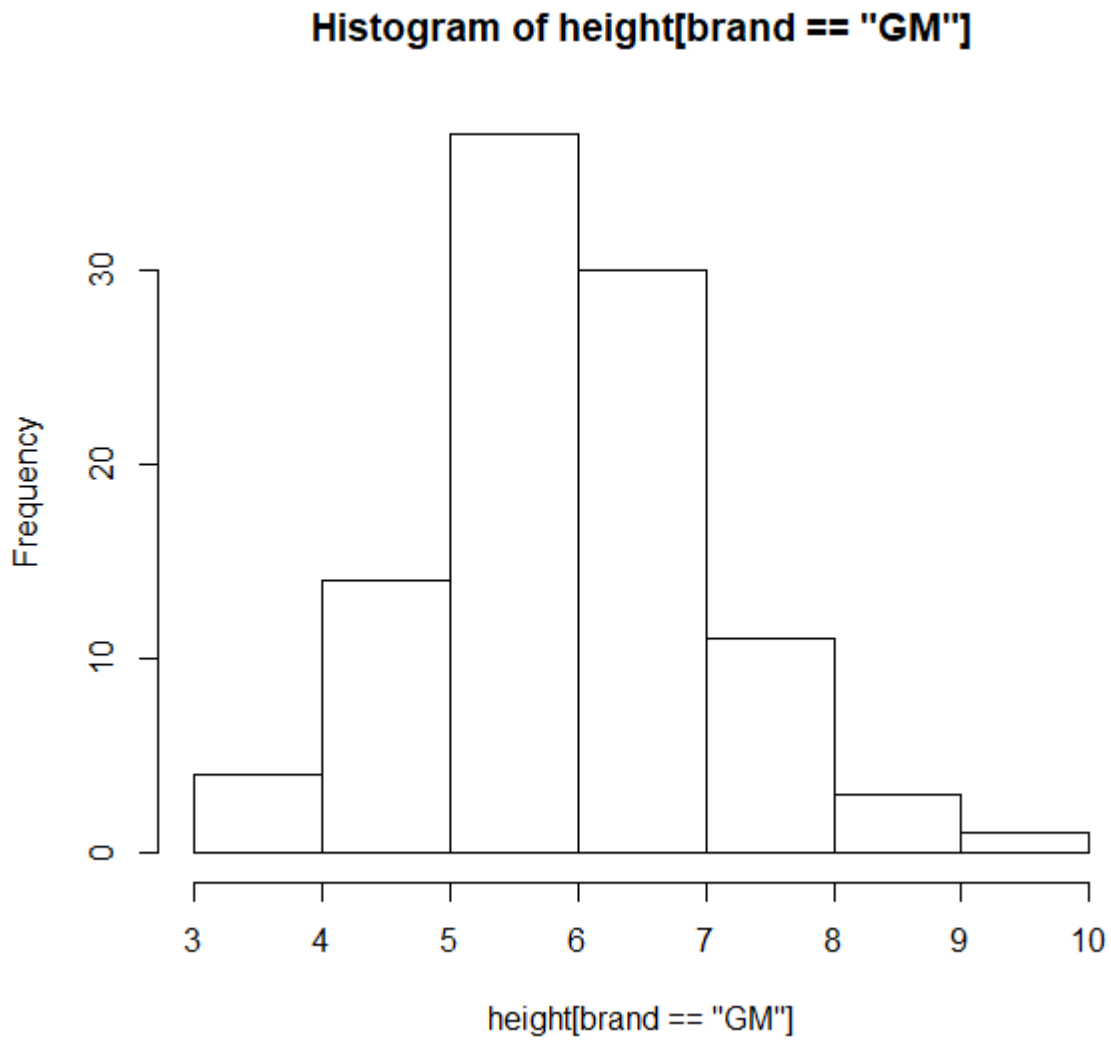
```
81  hist(height[brand=="Ford"])
```



Harshith Bondada
M10724854

This represents the histogram for height for GM brand car.

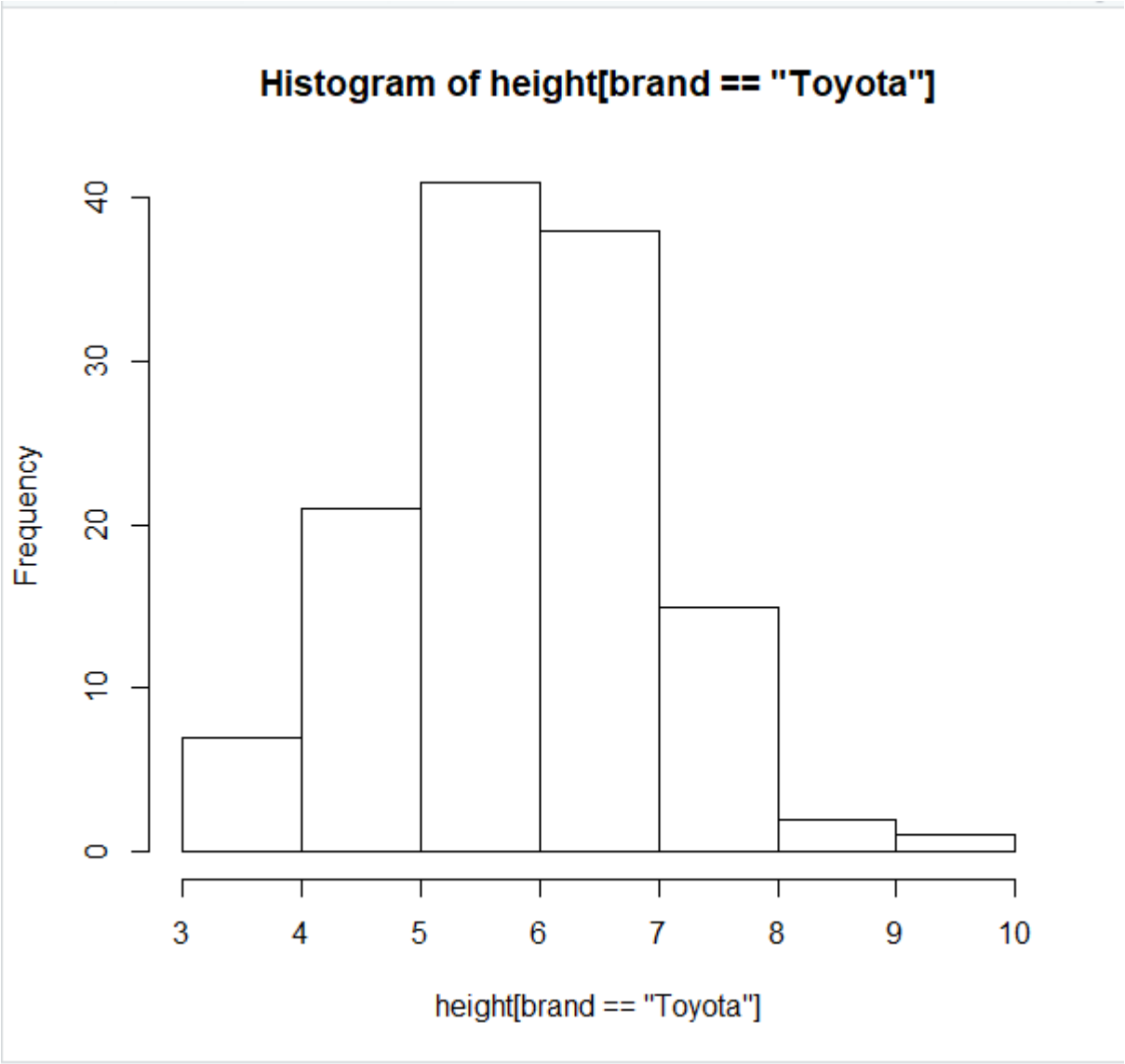
```
81  hist(height[brand=="GM"])
```



Harshith Bondada
M10724854

This represents the histogram for height for Toyota brand car.

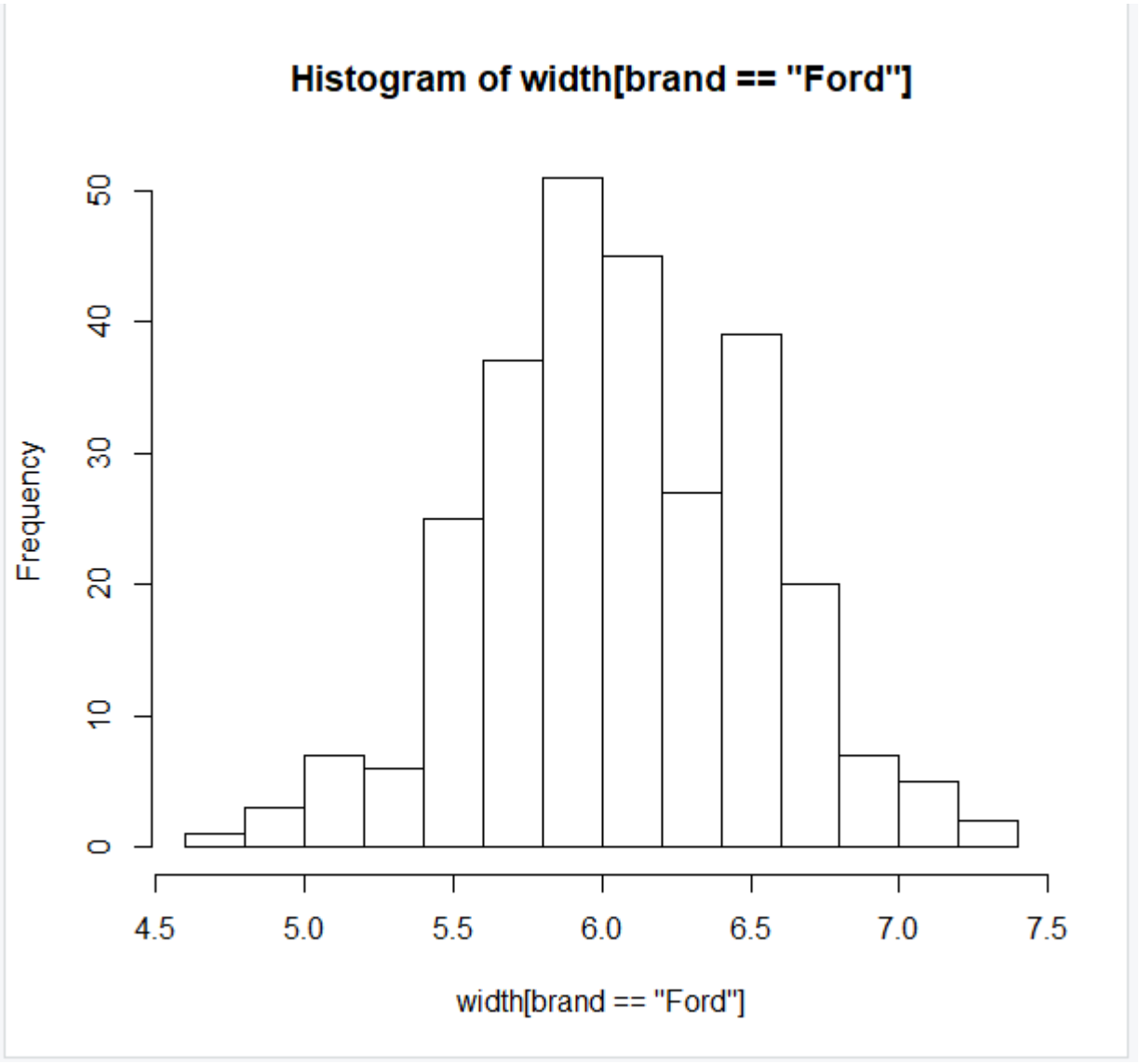
```
81  hist(height[brand=="Toyota"])
```



Harshith Bondada
M10724854

This represents the histogram for width for Ford brand car.

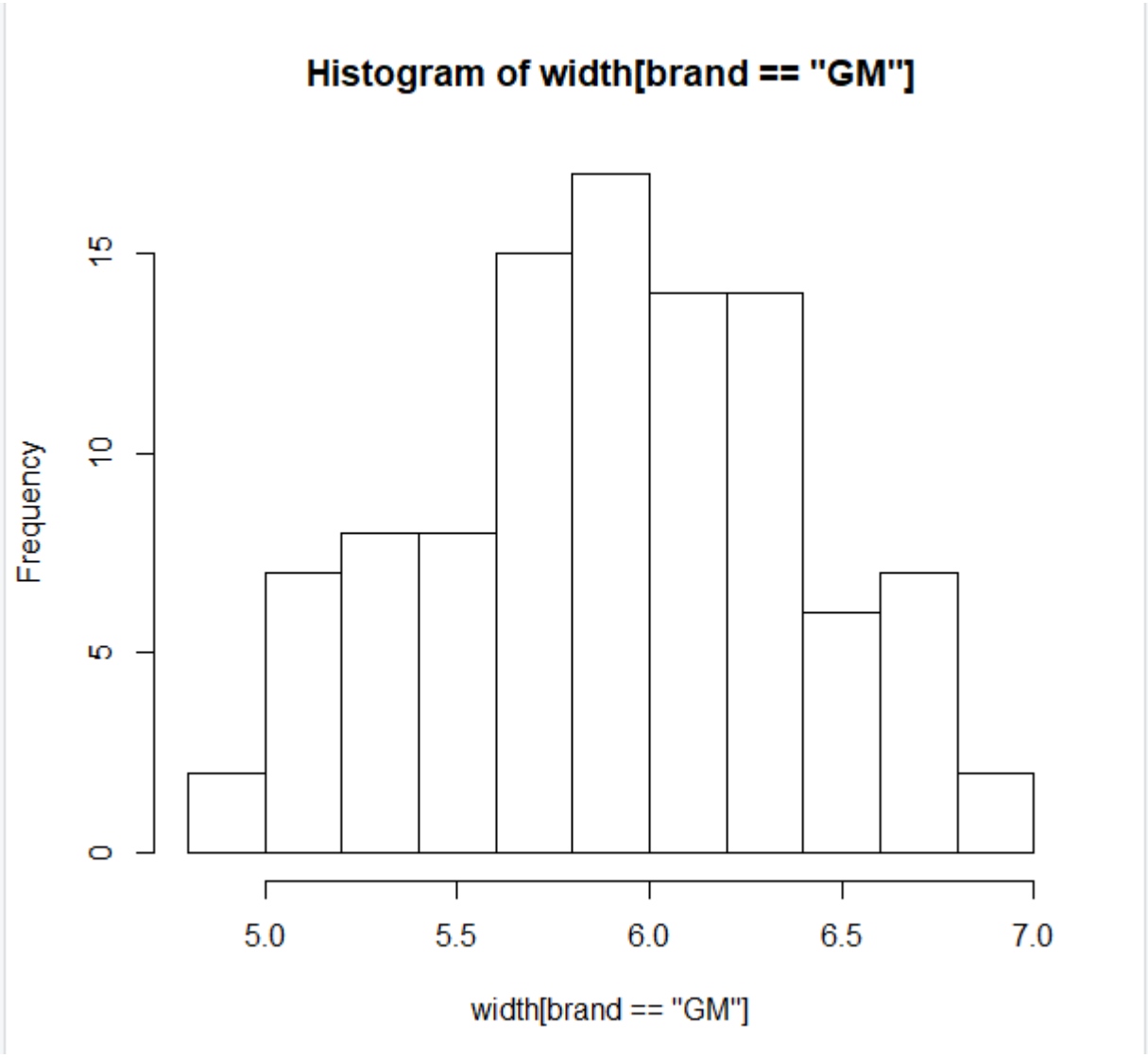
```
81  hist(width[brand=="Ford"])
```



Harshith Bondada
M10724854

This represents the histogram for width for GM brand car.

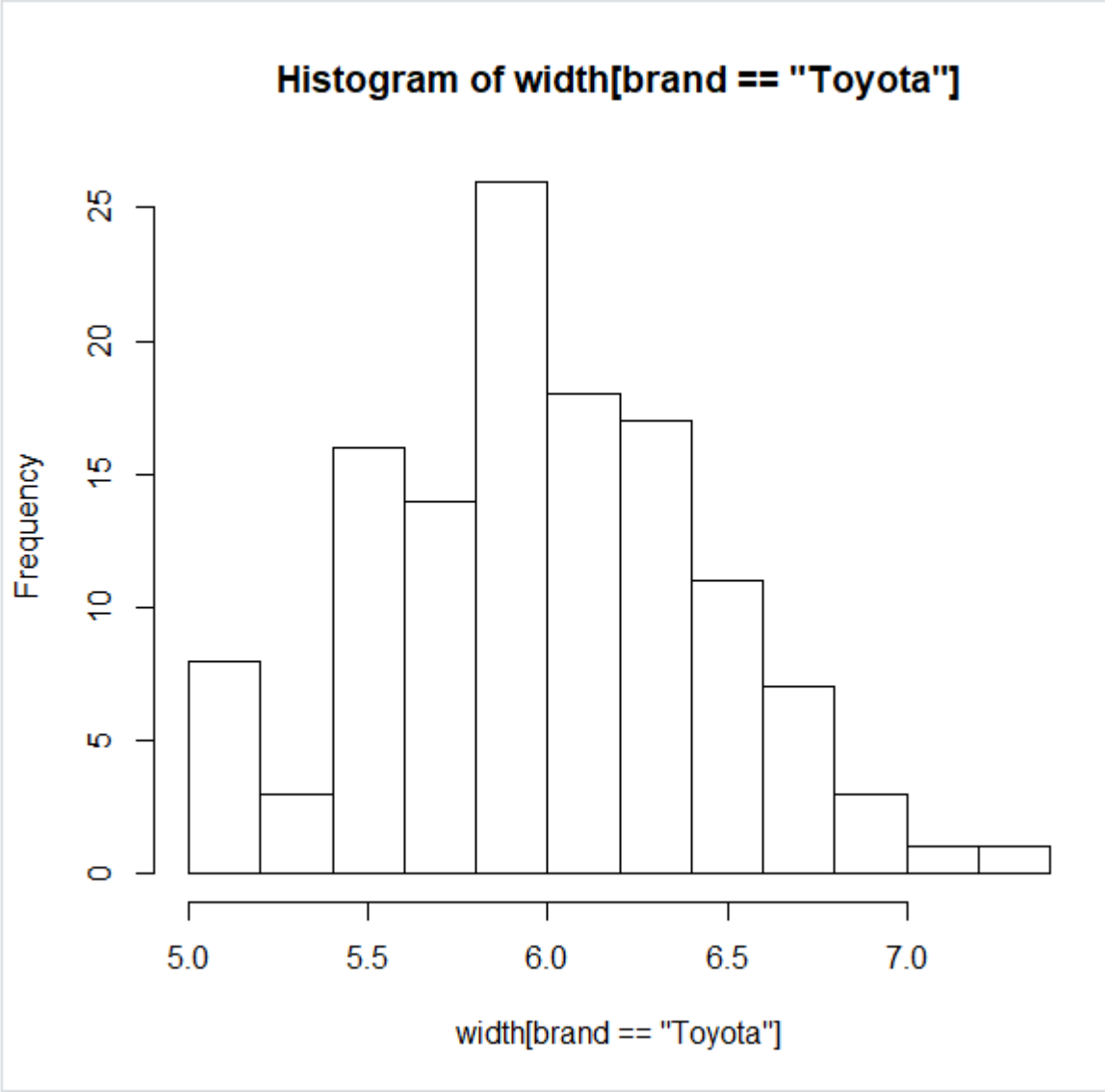
```
80  sd(ABS[brand== "Toyota"])
81  hist(width[brand=="GM"])
82
```



Harshith Bondada
M10724854

This represents the histogram for width for Toyota brand car.

```
81  hist(width[brand=="Toyota"])
```



Harshith Bondada
M10724854

6. Is there any missing value in the data set? If yes, which variable? What is the proportion of missing values?

The variable "mileage" has missing values in the data set.

This can be found out by using the sum function and in that by using the generic function **is.na**

```
> sum(is.na(home_work_1$mileage))
[1] 100
> sum(is.na(home_work_1$num_accidents))
[1] 0
> sum(is.na(home_work_1$num_passengers))
[1] 0
> sum(is.na(home_work_1$speed_car))
[1] 0
> sum(is.na(home_work_1$speed_air))
[1] 0
> sum(is.na(home_work_1$height))
[1] 0
> sum(is.na(home_work_1$width))
[1] 0
> sum(is.na(home_work_1$ABS))
[1] 0
```

We us the command sum(is.na(home_work$mileage) to get the count of the NA's in the variable mileage.We find that there are 100 NA's in the variable mileage.

```
83  sum(is.na(home_work_1$mileage))/length(home_work_1$mileage)
```

We divide it with the length function to get the proportion of the missing values.

Here, the length function only gets the length of the variable mileage in home_work_1.

```
> sum(is.na(home_work_1$mileage))/length(home_work_1$mileage)
[1] 0.2
```

Harshith Bondada
M10724854

7. Calculate the relative speed of the car (defined as = speed_car + speed_air, where speed_car is always positive and speed_air can be positive or negative). What is the average relative speed of the car? Convert speed_air to absolute value and calculate the average of the absolute value of speed of the air?

We use the following command to get a new column of relative_speed in the dataset.

home_Work<- transform (home_work_1, relative_speed = speed_car + speed_air)

Here, the transform function converts it'sarguments to a new data frame(**home_Work**)

We then calculate the relative speed of the car using the arithmetic operation.

```
84  ?transform
85  home_work <- transform(home_work_1, relative_speed = speed_car + speed_air)
```

Harshith Bondada
M10724854

You can see the relative_speed of the cars in the screenshot below.

Note: For space reasons I am only taking the screen shot up to 341 observation only

```
> relative_speed
  [1] 55.97915 35.42568 38.92249 35.03104 49.15909 59.76762 63.77775 38.41737 54.53329 33.03650
 [11] 40.32372 72.71861 55.70182 59.68569 55.24816 67.85296 38.70944 31.64759 57.54185 49.99437
 [21] 30.62349 62.18387 70.84696 57.58740 39.08617 37.22871 24.65116 73.94879 42.73822 45.12369
 [31] 41.18704 72.47228 32.30038 43.15835 56.21040 48.57623 36.07463 67.51163 64.12629 47.93282
 [41] 39.68407 52.75839 32.53616 35.54978 36.01789 33.13098 31.13688 50.05023 47.20485 53.35476
 [51] 49.53118 53.67068 36.02733 54.83844 44.53770 45.72005 31.23218 43.18729 64.14390 57.52719
 [61] 40.57174 55.09371 49.37332 49.04128 53.55516 47.71083 45.24597 54.26958 73.62997 49.96982
 [71] 52.45604 47.46031 54.98122 63.19409 48.64985 41.78041 52.46589 47.79416 53.36262 48.50421
 [81] 40.31239 63.96782 35.61630 64.09035 49.02052 33.28303 53.79559 70.55933 59.24124 72.80036
 [91] 42.93425 40.25147 36.20673 56.85474 41.65001 51.82736 49.27096 41.04586 66.56981 62.81555
[101] 45.24244 57.30096 36.51100 38.46083 60.88251 72.28408 59.66072 50.38595 50.54439 54.98874
[111] 54.44558 69.97130 43.15366 63.38026 50.95226 42.44716 63.13574 61.33945 42.34745 58.83554
[121] 62.22383 54.79156 56.26415 59.30731 40.78056 52.13707 70.84769 56.47870 39.60479 44.06753
[131] 54.83611 50.71982 36.77196 52.95458 45.76313 59.05715 45.98924 51.47130 51.91587 63.50617
[141] 47.10440 61.48969 59.26885 39.57681 42.12753 44.05209 53.79359 42.55408 38.32067 67.76869
[151] 37.77871 46.79350 50.50913 57.63231 58.95161 50.86065 50.48001 61.72406 43.29876 56.12479
[161] 32.97769 46.32484 59.07587 47.51480 61.13101 52.96852 47.28479 44.63251 69.33259 60.08981
[171] 48.83945 52.45252 41.52164 34.65058 37.34088 60.84711 78.85304 54.03086 42.06416 43.16668
[181] 42.28500 56.24944 39.70640 72.93865 41.63453 44.33016 41.03874 59.29069 65.54426 36.56413
[191] 43.93946 52.37904 33.82362 52.34402 60.80083 58.08786 59.05445 48.71950 61.33781 60.99592
[201] 63.25204 41.71891 40.09942 46.49023 61.71038 43.00215 51.59725 55.97129 40.11054 43.47237
[211] 49.10337 47.53437 47.25439 54.74061 64.81583 33.69793 63.96484 43.66881 50.01196 52.77275
[221] 54.40306 35.94013 54.44065 41.17901 47.63552 36.90863 36.39120 55.46782 48.83961 64.78257
[231] 22.44792 52.38577 58.83214 49.75577 59.77734 55.70442 40.16853 49.96577 41.47604 48.64759
[241] 60.05666 62.19096 48.99048 52.32650 42.57048 41.09128 38.60511 18.13336 50.89988 51.96107
[251] 58.12922 53.31355 47.18310 61.91986 19.52300 45.64708 60.15744 61.67043 42.04754 61.39205
[261] 68.92397 39.94795 48.13757 67.31628 68.34802 43.75491 34.81430 61.95592 55.78865 40.25353
[271] 51.56545 46.77616 51.54116 79.34006 57.71865 47.30262 45.93651 51.98797 67.98356 49.97561
[281] 41.52977 63.08108 46.69887 66.64924 58.24931 57.38576 64.66808 56.39597 46.45227 63.57890
[291] 47.82185 54.49771 37.73299 50.55026 47.44846 43.96552 64.62864 51.51512 40.99680 27.68674
[301] 51.48233 56.19672 40.61914 47.16713 41.07812 38.16643 47.75330 43.75030 53.58205 52.78430
[311] 51.28244 31.10688 60.25271 47.92702 52.78637 38.08669 72.67631 43.81702 23.66811 52.08756
[321] 55.50677 52.08049 39.56384 62.02716 40.18271 42.89091 67.97341 66.99718 39.51675 35.16574
[331] 41.98755 42.54363 46.05076 43.30495 40.88459 39.08929 41.83724 51.81867 41.75070 44.55912
[341] 48.59256 43.85250 47.16341 44.83575 51.24415 55.73398 56.59025 37.33691 51.51826 61.59852
```

We now calculate the averagerelative_speed of the cars using the below command.

I am selecting to calculate average for only relative_speed in the dataset by using the $ symbol.

```
92  mean(home_work$relative_speed)
```

The average relative speed of the cars is 50.30512

Harshith Bondada
M10724854

```
> mean(home_work$relative_speed)
[1] 50.30512
```

We now convert the speed_air to the absolute_value.

We type the following command:

```
93  Home_Work <- transform(home_Work, absolute_value_speed_air=abs(home_work$speed_air))
```

We have created a new variable called absolute_value_speed.

abs ()function converts the speed_air into their absolute values. I have used the transform function again and masked the all variable names of home_Work to Home_work expect absolute_value_speed_air which we created.

We now calculate the average of the absolute speed_air using the mean function.

```
> mean(Home_Work$absolute_value_speed_air)
[1] 2.508476
```

8. How many cars have mileage less than 40000? How many cars have height less than 5? Please delete those observations (i.e., cars whose mileages are less than 40000 and cars whose heights are less than 5) and delete the observations that contain NAsfrom the original data set to form a new data set.

We use the following command to obtain the number of cars having mileage less than 40000 and height less than 5.

less_mileage vector has the cars having mileage less than 40000

less_height vector has the cars having height less than 5.

```
18  less_mileage=which(Homework$mileage<40000)
19  less_height=which(Homework$height<5)
```

We use the length function to obtain the number of cars who met the above conditions.

We 210 cars for less_mileage and 93 cars have height less than 5.

```
> length(less_mileage)
[1] 210
> length(less_height)
[1] 93
```

We type the command in line 119 to delete the observations with mileage less than 40000 and observations with height less than 5. We use the OR operator to carry the logic.

In the line 120, we use the funtion**na.omit()** to omit all the NA observations from the dataset.

```
119 Home_Work<-Home_Work[!(Home_Work$mileage<40000 | Home_Work$height<5),]
120 Home_Work=na.omit(Home_Work)
```

We get the output upon successfully executing the commands.

```
> Home_Work<-Home_Work[!(Home_Work$mileage<40000 | Home_Work$height<5),]
```

```
> Home_Work=na.omit(Home_Work)
```

9. Divide the new data set (as obtained in Step 8) into three subsets: Ford, GM and Toyota.

We type the following commands to create a new subset for FORD, GM, TOYOTA

The **which** function returns the array indices and stores the FORD, GM, TOYOTA data in their respective subsets.

```
124 Ford=Home_Work[ which(Home_Work$brand=='Ford'), ]
125 GM=Home_Work[ which(Home_Work$brand=='GM'), ]
126 Toyota=Home_Work[ which(Home_Work$brand=='Toyota'), ]
127 dim(Ford)
128 dim(GM)
129 dim(Toyota)
```

The output is as follows:

Harshith Bondada
M10724854

New subsets were created and when we find out the dimensions of new subsets we get the number of observations across their variables.

Note:
Two new variables were added, Relative_speed and Absolute value for the speed_air

Ford data set has 86 observations over 11 variables.

GM has 38 observations over 11 variables.

Toyota has 34 observations over 11 variables.

```
> Ford=Home_Work[ which(Home_Work$brand=='Ford'), ]
> GM=Home_Work[ which(Home_Work$brand=='GM'), ]
> Toyota=Home_Work[ which(Home_Work$brand=='Toyota'), ]
> dim(Ford)
[1] 86 11
> dim(GM)
[1] 38 11
> dim(Toyota)
[1] 34 11
```

We can also use the Subset Function and create a sub dataset:

Note:

I am using the old data set, there would be only 9 variable parameters. ie, no relative_speed and absolute_value.

I am just using this to show that the work can also be done using the subset function.

The subset function creates a subset of Ford_dataset from the original Home_work dataset.

We are using the dim function just to show how many observations were formed a subset from the original dataset.

```
149 Ford_dataset = subset(Home_Work,brand=='Ford')
150 dim(Ford_dataset)
151 GM_dataset = subset(Home_Work,brand=='GM')
152 dim(GM_dataset)
153 Toyota_dataset = subset(Home_Work,brand=='Toyota')
154 dim(Toyota_dataset)
```

```
> Ford_dataset = subset(Home_Work,brand=='Ford')
> dim(Ford_dataset)
[1] 86  9
> GM_dataset = subset(Home_Work,brand=='GM')
> dim(GM_dataset)
[1] 38  9
> Toyota_dataset = subset(Home_Work,brand=='Toyota')
> dim(Toyota_dataset)
[1] 34  9
```

10. Using the new data set (as obtained in Step 8), is there any difference between these three brands (in terms of speed, height, width)? You can compare their means, variances.

We now try to compare the mean, variance in mileage, speed_car, height, width and also we compare ABS in each of the subsets i.e., whether they are present are not.

We use the table function to compare the ABS i.e., in how many cases they are present and in how many cases they are not present.

```
131 mean(Ford$speed_car)
132 mean(GM$speed_car)
133 mean(Toyota$speed_car)
134 var(Ford$speed_car)
135 var(GM$speed_car)
136 var(Toyota$speed_car)
137 mean(Ford$height)
138 mean(GM$height)
139 mean(Toyota$height)
140 mean(Ford$width)
141 mean(GM$width)
142 mean(Toyota$width)
143 mean(Ford$mileage)
144 mean(GM$mileage)
145 mean(Toyota$mileage)
146 table(Ford$ABS)
147 table(GM$ABS)
148 table(Toyota$ABS)
```

Harshith Bondada
M10724854

We see that Toyota cars move with greater speed on Avg than Ford and GM.

Toyota cars dominate in height whereas, Ford dominates the other two in width.

Ford cars have the greater mileage when compared to the other two cars.

```
> mean(Ford$speed_car)
[1] 51.43112
> mean(GM$speed_car)
[1] 52.23822
> mean(Toyota$speed_car)
[1] 52.25184
> var(Ford$speed_car)
[1] 93.43505
> var(GM$speed_car)
[1] 77.06144
> var(Toyota$speed_car)
[1] 97.62885
> mean(Ford$height)
[1] 6.313565
> mean(GM$height)
[1] 6.25896
> mean(Toyota$height)
[1] 6.440077
> mean(Ford$width)
[1] 6.056579
> mean(GM$width)
[1] 5.932025
> mean(Toyota$width)
[1] 6.048129
> mean(Ford$mileage)
[1] 48788.14
> mean(GM$mileage)
[1] 48324.16
> mean(Toyota$mileage)
[1] 48482.2
```

Harshith Bondada
M10724854

This below screen shot the comparisons for ABS.

We find that ABS are present Ford cars with highest count of 52 while, GM and Toyota have same count for ABS despite, a slight variation in the total count.

```
> table(Ford$ABS)

FALSE  TRUE
   34    52
> table(GM$ABS)

FALSE  TRUE
   21    17
> table(Toyota$ABS)

FALSE  TRUE
   17    17
```